# Reduced order models for geometrically parameterized beams and shells

**Luís Guilherme Simões da Silva Tão Espírito Santo**

Dissertation to obtain the Master of Science Degree in

## Civil Engineering

Supervisor(s): Prof. Doctor Pedro Diez Mejía
Prof. Doctor José Paulo Baptista Moitinho de Almeida

## Examination Committee

Chairperson: Prof. Doctor Orlando José Barreiros D'Almeida Pereira
Supervisor: Prof. Doctor José Paulo Baptista Moitinho de Almeida
Member of the Committee: Prof. Doctor Sergio Zlotnik Martinez

**October 2022**

# Statement

I declare that this document is an original work of my own and that it meets all the requirements of both University of Lisbon and Universitat Politècnica de Catalunya · Barcelona Tech - UPC Code of Conduct and Good Practices.

# Acknowledgments

First and foremost, I must thank my supervisors. To Professor Pedro Diez from UPC, Universitat Politècnica de Catalunya, for the opportunity to develop this academic project in Barcelona, for his unconditional guidance and trust. To Professor José Paulo Moitinho from IST for his assistance and dedicated involvement in every step throughout the process, for reviewing my progression and cosmic patience.

Finally, but most importantly, I would like to thank my family and girlfriend for whom no words can describe all the value that their support and care has for me. To my parents for their constant carefulness, presence, for being role models. For my girlfriend, Nicole, for being there for me, for the back and forth trips between Rio de Janeiro and Barcelona, for the companionship, for the love. I would also like to congratulate everyone who helped me build the person I am, for the teaching. To my friends.

# Resumo

Recentes avanços em programas de modelação e análise estrutural reintroduziram o interesse em coberturas de casca fina, que se destacam pela sua eficiência e estética. A forma de uma casca pode ser avaliada através do seu coeficiente de flexão, quociente entre a energia de flexão e a energia total de deformação. Para uma casca extensivamente multiparametrizada, métodos numéricos clássicos podem desencadear a maldição da dimensionalidade, já que uma solução é calculada para cada conjunto de parâmetros. Assim, métodos alternativos, como Modelos de Ordem Reduzida (*Reduced Order Model* - ROM), tornam-se relevantes neste contexto.

A presente dissertação foca-se em encontrar formas antifuniculares para cascas geometricamente parametrizadas. A Decomposicão Generalizada Adequada (*Proper Generalized Decomposition* - PGD) é aplicada para obter uma solução paramétrica explícita, facilitando a minimização do coeficiente de flexão da estrutura. O ROM baseou-se num Modelo de Ordem Completa (*Full Order Model* - FOM), tomado como referência.

O modelo geométrico é definido até três parâmetros, com a formulação das Superfícies de Bézier. O modelo mecânico é baseado na formulação de Mindlin-Reissner e analisado numericamente com uma malha triangular com aproximações de segunda ordem.

Os resultados demonstram que a nova aplicação de PGD, para estruturas de casca parametrizadas, permite encontrar soluções antifuniculares. O ROM é uma boa aproximação do FOM para vários desígnios práticos, não apenas para otimização com ganhos significativos em tempo e memória computacional. Visualizar a informação de uma configuração paramétrica multidimensional pode ser um desafio, assim uma Interface Gráfica do Utilizador (*Graphical User Interface* - GUI) foi desenvolvida especificamente para pós-processamento da solução PGD, exibindo as soluções paramétricas em tempo real.

**Palavras-chave:** Cascas finas, Descoberta de Formas, Antifunicular, Modelo de Ordem Reduzida, Decomposição Generalizada Adequada (PGD)

# Abstract

Reawakened by modern advances in structural design and modelling software, thin-shell covers certainly stand out due to their efficiency and aesthetics. The form suitability of an arbitrary thin-shell can be measured by its bending coefficient, the ratio between the bending and total strain energies. For an extensively multi-parameterized shell, classical numerical methods may invoke the curse of dimensionality, as one solution must be computed for each set of parameters. Thus, alternative methods, such as Reduced Order Methods (ROM), are relevant in this context.

The present work focuses on finding antifunicular forms for geometrically parameterized shells. Proper Generalized Decomposition is used to obtain an explicit parametric solution. Thus, minimization of the structure´s bending coefficient is easily performed. The ROM relied on an accurate Full Order Model (FOM) which is taken as reference.

The geometrical model is defined up to three parameters, with the Bézier Surface formulation. The mechanical model is based on the Mindlin-Reissner formulation, numerically analyzed with a triangular mesh, with quadratic approximations.

The results demonstrate that the novel PGD application to parameterized shell structures is successfully allowing to find the antifunicular forms. The ROM is a fair approximation of the FOM for many practical purposes, not only for optimization, with significant gains in computational time and memory resources. A crucial issue in a multidimensional parametric setup is its visualization and, in general, the display of information. A Graphical User Interface (GUI) is specifically designed to postprocess the PGD solution, displaying the parametric responses in real-time.

**Keywords:** Thin-Shell Structures, Form-Finding, Antifunicular, Reduced Order Model, Proper Generalized Decomposition

# Contents

# List of Tables

# List of Figures

# Nomenclature

| | |
|---|---|
| FOM | Full Order Model. |
| GUI | Graphical User Interface. |
| PGD | Proper Generalized Decomposition. |
| ROM | Reduced Order Model. |

**Greek symbols**

| | |
|---|---|
| $\epsilon^M$ | Relative error of an approximation. |
| $\eta$ | Stopping criteria coefficient. |
| $\gamma$ | Dimension assumed to be unknown. |
| $\nu$ | Poisson´s ratio. |
| $\Omega$ | Parametric Space. |
| $\Phi$ | Stress function. |
| $\rho$ | Density. |
| $\sigma$ | Amplitude of the mode. |
| $\sigma$ | Normal stress. |
| $\tau$ | Shear stress. |

**Roman symbols**

| | |
|---|---|
| $A$ | Area. |
| $\mathbf{B_C}$ | Bending Coefficient tensor. |
| $B_C$ | Bending Coefficient scalar. |
| $\mathbf{D}$ | Material property operator. |
| $D$ | Number of parameters. |
| D | Differential operator. |

| | |
|---|---|
| $E$ | Young´s Modulus. |
| $f$ | Parametric function. |
| $\mathbf{f}$ | Nodal force tensor. |
| $\mathrm{f}$ | Nodal force vector. |
| $g$ | Gravitational acceleration. |
| $h$ | Maximum height of the shell. |
| $\mathbf{K}$ | Stiffness tensor. |
| $\mathbf{k}$ | Stiffness vector. |
| $K$ | Stiffness matrix. |
| $L$ | Shell´s span. |
| $\mathbf{M_C}$ | Membrane Coefficient tensor. |
| $m$ | Moment per unit of length. |
| $M_C$ | Membrane Coefficient scalar. |
| $N$ | Stress per unit of length. |
| $n_i$ | Number of elements in $\Omega_i$. |
| $\bar{p}_z$ | Weight per unit of area. |
| $\mathbf{P}$ | Shell´s Bézier control points matrix. |
| $\mathbf{S_C}$ | Shear Coefficient tensor. |
| $S_C$ | Shear Coefficient scalar. |
| $t$ | Shell´s thickness. |
| $\mathbf{u}$ | Nodal displacement tensor. |
| $\mathcal{U}$ | Strain energy. |
| $u$ | Nodal displacement vector. |
| $x_i^j$ | Element $j$ on parametric space $i$. |
| $z_1$ | scalar associated with the z component of the center´s control point. |
| $z_2$ | scalar associated with the z component of the border´s middle control point. |
| $z_3$ | scalar associated with the z component of the diagonal´s second control point. |
| $z_4$ | scalar associated with the z component of the border´s quarter control point. |

| | |
|---|---|
| $z_5$ | scalar associated with the z component of the middle line´s second control point. |

**Subscripts**

| | |
|---|---|
| $b, m, s$ | Stiffness matrix components. |
| $i$ | Parametrical space index. |
| $j$ | Arbitrary stiffness matrix component. |
| $x, y, z$ | Cartesian components. |

**Superscripts**

| | |
|---|---|
| $\sim$ | PGD separated approximation. |
| $j_i$ | Element in sectional domain $\Omega_i$ |
| * | Adjoint. |
| - | Mean value. |
| M | PGD separation maximum mode. |
| m | Arbitrary mode. |
| T | Transpose. |

# Chapter 1

# Introduction

## 1.1 Motivation

In civil engineering, a type of structure that certainly stands out due to their elegance, beauty, and impact, are shells, namely thin-shell roof covers. From Los Manantiales in Mexico, Figure 1.1, Tenerife´s Opera House, Figure 1.2, Portugal´s Pavilion, Figure 1.3, Singapore´s Jewel Changi Airport, Figure 1.4, Lotus‘ temple in India, Figure 1.5 and Kresge´s Auditorium on MIT, Figure 1.6, one cannot deny that these masterpieces defied and continue to question common practices in architecture and engineering. Modern advances in software related to structural design and architectural modelling drastically simplified the process of bringing to life complex and abstract geometries, also known as free-form geometries. Thus, reawakening the interest in thin shell structures.

As the efforts to deliver structurally efficient and economical solutions without compromising the aesthetic component are continuously increasing, the application of thin-shell structures to covers becomes even more suitable. Thin shell structures can cover enormous spans without significant stresses, while minimizing material consumed by mobilizing the majority of the cross section´s resistant capacity, when properly designed.

Felix Candela, a renowned architect and master builder of thin shell structures in the 20th Century, stated: "If possible, bending stresses need to be avoided, through the selection of an appropriate shape" [1]. To determine a suitable shape, or optimal shape, for a given problem, one must find the geometry that minimizes the bending energy, and maximizes the membrane energy. However, estimating the optimal shape can be a rather costly and arduous exercise regarding computer memory or/and time invested, even with the most advanced software.

The analysis of a given shape form will depend on various variables, such as the surface points, material properties, and boundary conditions. Therefore, for each combination of variables, a geometry has to be generated and a structural analysis has to be carried out. As the number of variables increases linearly, the number of simulations needed to analyze the problem increases exponentially. To illustrate the seriousness of the problem, imagine the example taken from [2]. For an arbitrary model, D = 30 parameters, with a discretization M = 1000, the number of simulations needed to evaluate the whole system, $M^D = 1000^{30}$ , would top the number of particles in the Universe, [2].

Therefore, more efficient and faster methods such as Reduced Order Models, ROM, were developed to ease and simplify the process of solving and analyzing complex systems. One of those methods is the PGD, Proper Generalized Decomposition. This method provides explicit parametric solutions, denoted as computational vademecums or digital abacuses, to parametric boundary value problems [3]. The solution is provided as the sum of $N$ modes, where each mode is the product of $D$ sized $M$ vectors. That is to say, instead of storing $1000^{30}$ elements, the PGD solution will provide an approximate solutions with only $1000 \times 30 \times N$ elements. Thus, the complexity of the problem increases linearly with the number of variables, generally, reducing drastically the computational costs and/or time consumed to estimate those solutions.

This works will focus on applying the PGD to estimate the anti-funicular shape, for a given set of variables that characterize a thin shell roof cover, and assess the relevance of the method in this application.

Figure 1.1: Los Manantiales (Félix Candela,1958)

Figure 1.2: Tenerife´s Opera (Santiago Calatrava,2003)

Figure 1.3: Portugal´s Pavillon (Alvaro Siza Vieira 2010)

Figure 1.4: Jewel Airport (Moshe Safdie 2019)

Figure 1.5: Lotus Temple (Fariborz Sahba 1986)

Figure 1.6: Kerge Auditorium (Eero Saarinen, 1955)

## 1.2 Objectives and Deliverables

The goal of this dissertation is to conceive a Reduced Order Model, ROM, with the PGD approach capable of finding anti-funicular shapes for thin-shells structures, through the estimation of the parameters that minimize the bending coefficient, $B_C$.

Form finding techniques and surface generation formulations will be evaluated to choose the best path to parameterize and generate the shell´s geometry. Membrane theory will be studied to acquire a tool to validate the generated models. The PGD routines in [3] will be used to create the ROM. The FEM routines displayed in [4] will be optimized and adjusted to solve hundreds of parameterized shell structures, these results will be taken as reference. An overview of the history of shell design will also be presented to help assess results in a practical fashion.

All in all, the biggest outcome of this dissertation will be to obtain a general idea about thin-shell structures: from achieving a fair PGD approximation, to find antifunicular shape and perform their analysis. Aside of successfully creating the Reduced Order Model, it will be interesting to have a clear perception of the necessary improvements to broaden the PGD application in this field.

## 1.3 Dissertation Outline

The dissertation is divided into 7 Chapters, chosen and organized to give a general idea about thin-shell structures, form-finding techniques, shell structural analysis, the Proper Generalized Decomposition and, ultimately, be able to generate an accurate FOM and reproduce its ROM counterpart. Therefore, the literature review will be presented throughout the first four chapters, whereas, the methodology, results, discussion, and future works be extensively detailed along Chapters 5, 6 and 7.

The current chapter, Chapter 1, serves, mainly, to set the motivation for this dissertation, express its objectives and organization.

In Chapter 2, an overview of shell design history, shell classification, advantages and disadvantages of shell structures, and form-finding techniques that can be applied within the scope of this dissertation, is carried out.

In Chapter 3, a theoretical background on shell analytical analysis and membrane theory is presented. The aim of this chapter is to learn which forces are at stake, the limitations of this type of analysis and to attempt to draw a closed-form expression for membrane stresses, for an arbitrary analytically defined geometry. Hence, acquiring an adequate analytical tool to validate the Full Order Model.

In Chapter 4, a brief explanation of the PGD is introduced. Furthermore, an overview of the articles detailing the PGD approach in its algebraic version, [3] and [5], is presented.

In Chapter 5, the tools presented in previous chapters are agglutinated and the FOM is created and validated, capable of promptly generating various shapes of shells and delivering reliable results in terms of displacements, stresses, and energy coefficients. The PGD routines´ input is also defined. All the steps required to build and validate the models, and generate the results are also detailed in this chapter.

In Chapter 6, the steps to create the ROM are executed. The results provided by the ROM and FOM are compared in terms of relative error and memory usage, for one, two and three parameters. An assessment of all results is conducted, to ensure their accuracy and validity.

In Chapter 7, a conclusion is drawn, namely, in the novel PGD application, regarding its advantages and limitations. Several considerations are also presented concerning future developments and the benefits of the PGD in similar problems.

At last, the literature used in this dissertation will be presented.

# Chapter 2

# Shell Characterization

## 2.1 Shell Definition

Shells structures are present in an extensive domain of applications due to their structural efficiency. From the design of balloons or tennis balls to the design of more complex structures as cooling towers, roof-covers, pipes, water tanks, silos, car bodies, aircrafts or even missiles. Not only in engineering applications, but shell structures are also found in nature as leafs, cranes, eggs, bubbles, etc. [6]. In civil engineering, shell structures are mostly applied in industrial buildings or in special structures, for aesthetics and architectural purposes.

As shell structures can be applied on countless applications, its definition will be presented in a broad sense. A shell is any structure defined by one or more curved surfaces, whose thickness is always notably lower than the other two dimensions. The structural behaviour of the shell is highly dependent on its curvature. Shell structures resist loading through the combination of bending and membrane forces. Whereas the latter are more efficient, as they mobilize the resistant capacity of the entire cross section, the former can be necessary to avoid buckling phenomena and possible inextensible modes of deformation, [7]. For the sake of stress analysis, a shell can be defined as an object, which may be considered as the materialization of a curved surface [8].

## 2.2 A brief review of the history of thin shells and physical form finding

### 2.2.1 The Catenary - the true form of an arch

Only in 1675, Robert Hooke summarizes the ”true mathematical and mechanical form of all manner of arches for building” [9] through an anagram published in [9], corresponding to the following statement: ”As hangs the flexible line, so the inverted will stand the rigid arch” [10]. That is to say, when a chain is submitted to a certain loading, it will deform and assume a shape where equilibrium is guaranteed, in this case, in pure tension. Hence, if the shape is rigidified and reversed, the outcome will result in an arch, whose shape consists of a path, the thrust line, connecting both supports with only compressive forces. If pure compression is achieved, the antifunicular shape for a given load is obtained, as seen in Figure 2.1. Although Robert Hooke was unable to derive the catenary equation, that is, the analytical expression of a string with constant load, and constant cross-section, hanging freely from two points of suspension, [11], David Gregory proved the catenary was the correct shape and derived its analytical expression.

$$y(x) = -a \times \cosh \frac{x}{a} \tag{2.1}$$

It is worth to mention that, through the employment of the hanging chain principle, [9], Giovanni Poleni (1748) was able to assess the safety of St. Peter´s Dome, by dividing it in 32 arches and checking if a hanging chain, for a load proportional to the weight of the slice, would fit within the surfaces of the dome. By combining this principle with the introduction of the parallelogram rule by Simon Stevin (1548-1602), the foundations of Graphical Statics were set, only to be formalized by Culman in [12].

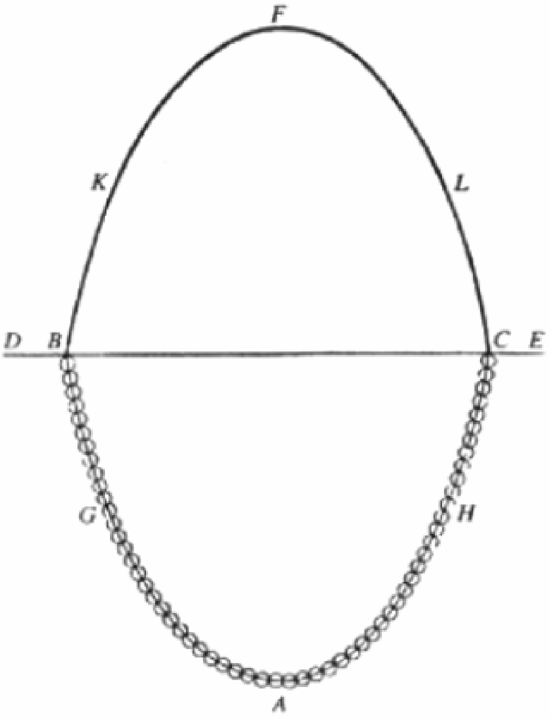


Figure 2.1: Inverted hanging chain principle, adapted from [13]

This breakthrough allowed engineers to complement experimental results from models with mathematical analysis and graphical methods, tying thrust lines with collapse loads. From thrust line theory, it is concluded, in parallel with David Gregory, that if a thrust line can be found within the structure´s boundaries then the structure can be shown to be safe for that set of loads [14]. The thrust line is defined as the geometrical locus of the points of application of sectional forces. In 1773 Couloumb also established that if an arch develops a certain number of hinges, points where the thrust line intersects the structure´s boundaries, the arch will collapse. Therefore, by manipulating the thrust line within the masonry thickness, maximum and minimum loads could be calculated during the project´s design phase.

All these concepts were used and generalized into three dimensions, allowing following architects and

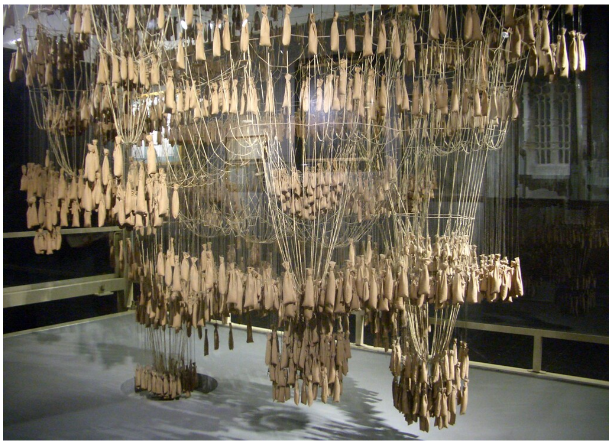


Figure 2.2: The hanging model of Colonia Guell

masterbuilders to achieve new anti-funicular shapes, while assuring structural safety. Architects such as Antoni Gaudí (1852–1926) for the Colònia Güell, see Figure 2.2 and Frei Otto (b.1925) for the Multihalle used to reverse the loads on a pure tension structure, by means of systems of ropes and chains, to obtain pure compression structures.

### 2.2.2 Modern form-finding techniques

In the 20th century, roof shells tended to be mainly based on geometries that could be analytically defined such as spheres, cylinders, ellipsoids, paraboloids and hyperbolic paraboloids, as they offered analytical solutions for structural analysis. Despite easing the design process, some of these geometries did not necessarily lead to structures with a dominant membrane response. Thus, extra elements such as edge beams, stiffeners and prestressing had to be introduced, [15], as seen in Kerge Auditorium (Eero Saarinen, 1955) and Sydney Opera House (Jørn Utzon, Peter Brian Hall, 1973).

Felix Candela divided shell roofs into two groups, proper shells, that avoid bending stresses and are doubly curved, and improper shells, which carry the load through significant bending stress [16]. He applied into his designs the hyperbolic paraboloid geometry, which contemplates double-curvature, allows for an arch-effect in two planes and provides more alternative paths for the internal forces to reach the supports. In his most renowned work, by possessing knowledge of structural behaviour Candela was able to calculate stresses with a similar efficacy similar to modern FEM analysis [1], see Figure 2.3, only by exploiting equilibrium. Therefore, by understanding the locations with greater stresses, the thickness of the problematic areas could be increased. Thus, providing aesthetically pleasing, functional and inexpensive (the hypar allows the use of straight formwork elements) structural solutions, as seen in Figure 1.1.

While Candela´s work was mostly focused on analytically defined geometries, Heinz Isler focused on extending the principal of the hanging chain. By hanging wet sheets of cloth, in cold environments, Isler was able to generate models of funicular geometries, as the cloth would be frozen and only mobilizing membrane stresses in the day after [16]. By changing the angle between the cloth´s fibers, the resulting shape would change as well. This ”adapted” hanging chain principal allowed Isler to not only generate ”endless” scenarios for funicular shapes, by slightly modifying the boundary conditions and anisotropy,

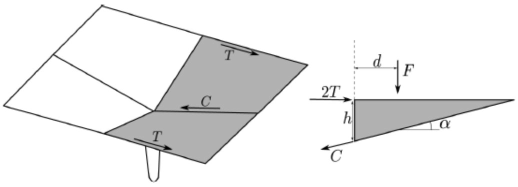
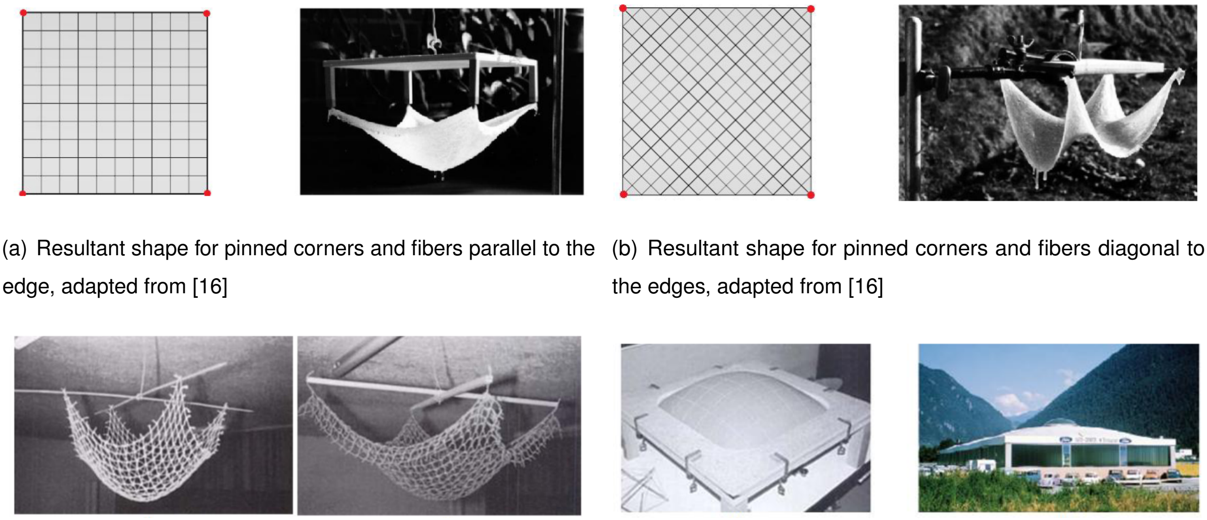


Figure 2.3: $C_{FEM} = 87,7kN$, $T_{FEM} = 40kN$ , $C_{EQ} = 87,9kN$, $T_{EQ} = 40.8kN$, Paraguas 1958, Félix Candela, adapted from [1]

but also see which fold provides better stiffening to the free edges. Air pressure was also used by the architect to inflate elastic membranes to generate funicular forms, as seen in Figure 2.4.



(a) Resultant shape for pinned corners and fibers parallel to the edge, adapted from [16]

(b) Resultant shape for pinned corners and fibers diagonal to the edges, adapted from [16]

(c) Effect of folding of the ”borders” on the final form, adapted from [16]

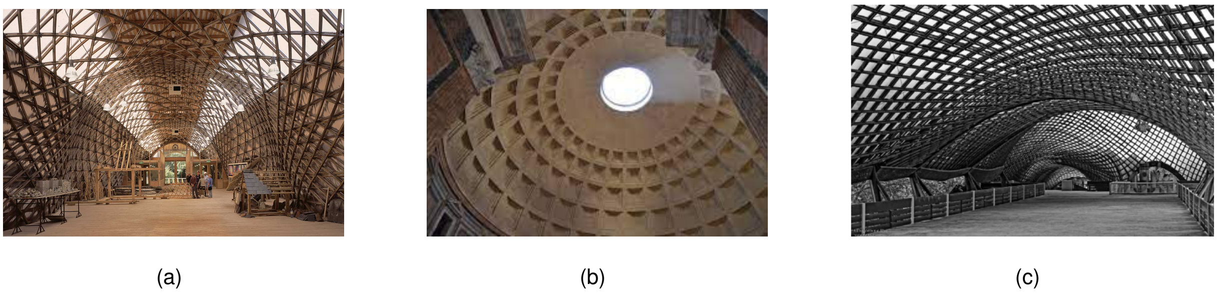
(d) Prototype and full scale model generated with the inflation technique, adapted from [16]

Figure 2.4: Form finding techniques introduced by Heinz Isler

### 2.2.3 Numerical form-finding

Physical form-finding continued to be praised, mostly by architects and experienced masterbuilders, following the guidelines mentioned in the previous Section. Nevertheless, the increasing computational power allowed for numerical form-finding methods, based on discretization of the hanging networking of chains into bars and nodes/particles and Hook´s principle, to execute virtual simulations for endless geometry combination, obtaining a CAD solution that can be instantly exported to structural analysis software. From all the methods, the most relevant approaches are the Stiffness Matrix Method SMM, the Force Density Method, FDM, the Thrust Network Analysis, TNA, the Dynamic Relaxation, DR, and the Particle-Spring System, PS. Generally, each of these methods is more suitable when applied to discover geometries in specific fields of application. The FDM is normally used to find timber grid shell forms, the TNA to assess the structural integrity of historical buildings, mainly unreinforced shell structures and the PS is adequate to generate thin shell concrete shells. The SMM is normally inadequate to directly generate a shell, as the simulation of a network of hanging chains implies large displacements, that can cause numerical instabilities and defy the assumption of small displacements, [17]. Not only that, but the

FDM and the DR methods are not generally effective at form-finding statically determinate structures that, under self-weight, act as pure membranes. Therefore, the only well-known method suited to solve the current problem is the PS. Independently of the chosen method, the following data has to be defined: nodal coordinates; fixed nodes; nodal topology; that is, connectivity of the system of chains; prescribed forces; and an iterative threshold.


(a)

(b)

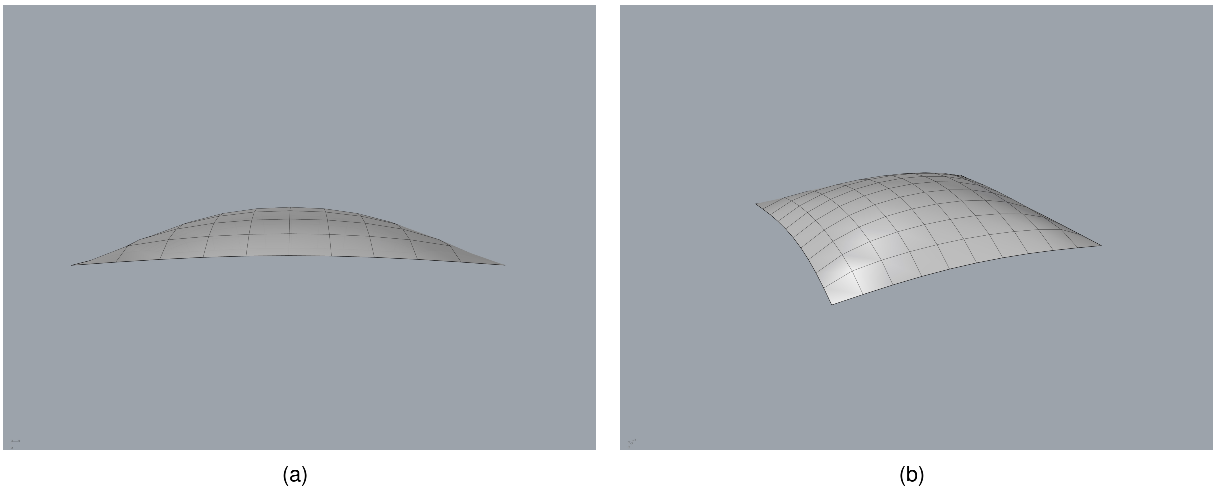
(c)

Figure 2.5: (a) Timber gridshell (The Weald and Downland Open Air Museum, Edward Cullinan Architects and Buro Happold, 2002) (b) unreinforced masonry shells (Rome Pantheon,125 d.C) (c) strained gridshells (Mannheim Multihalle, Frei Otto and Architects Carlfried Mutschler and Winfried Langner, 1974)

PS systems discretize a network of chains into nodes, or particles, that are connected by mass-less linear elastic springs. When displaced from their rest length, the springs will produce forces, due to their stiffness and length, which are previously assigned. As in the case of the 1D hanging chain, external forces can be applied on the particles, as due to gravitational acceleration. The PS system is generally not in equilibrium at the start of the simulation, thus, until equilibrium is reached, or until a certain level of equilibrium, set by a threshold, is not respected, the system will move, and each particle will have its own vector, regarding forces applied, position, velocity and acceleration. To prevent the system from purposelessly oscillating about the equilibrium configuration, damping coefficients are set when defining the springs´ properties. To numerically solve this problem, implicit solvers are more convenient, as small changes in the springs´ lengths cause the explicit approach not to converge. More insights about these methods may be found in [18].

Nevertheless, to explore the challenges of using this method on proper software, the shell in Figure 2.6 was designed on GrassHopper and rendered in Rhino using 121 particles, uniformly distributed on a span of $L = 10m$ in both directions, particles´ forces equal to $t \times \rho \times g/(11 \times 11) = 0.15 \times 2500 \times 10/(11 \times 11) = 30N$, being $t$ the thickness in meters, $\rho$ the mass density of the material and $g$ the rounded integer of the gravitational acceleration. The system has its corners pinned. The curvature of the final form was not taken into account on the calculation of the particles´ external forces and the code for this application can be found in Appendix A.1. The mesh was then exported to Matlab and structurally assessed. For this case, the membrane behaviour was not astonishing, the bending coefficient was around $0.20$ which is explained due to the low discretization in particles of the system, and high stiffness used, causing the height/span ratio to be low, 1.5, thus making the structure ”flat”.

However, it is costly to run simulations until a geometry with the desired behaviour is found. Picking up the necessary parameters, which are, essentially, the level of discretization, particle load and spring stiffnesses for a uniform distribution of points, a Full Order Model and its ROM counterpart could be created to execute this method and deliver the results through a black box. To dodge the trouble of



(a) (b)

Figure 2.6: Shell designed with the PR system on Grasshopper

interoperability between softwares, if the intent is to minimize bending energy, for a given parameterized surface formulation, one can take advantage of the FEM, and for a great variety of structures, choose the configuration that best behaves as a membrane, calculating the bending coefficients at once with the PGD. Thus swiftly obtaining a solution without any of the methods mentioned, for the potential cost of lack of diversity of form.

## 2.3 Classification of shells

### 2.3.1 Gaussian Curvature

The Gaussian curvature, k, is defined by the product of the principal curvatures, $k1 \times k2$. Infinite planes containing a vector normal to the shell´s middle surface, can be imagined for any shell. These planes are named normal sections and are also perpendicular of the middle surface. The intersection of each of those sections with the middle surface will result on the generation of infinite curves with different curvatures. The principal curvatures are defined as the curves with the highest, k1, and lowest, k2, curvatures, respectively. The Gaussian curvature can be positive, negative, or zero (simple curvature). If k$>$0, the curvature is classified as synclastic, anticlastic if k$<$0, or with single curvature if k$=$0.

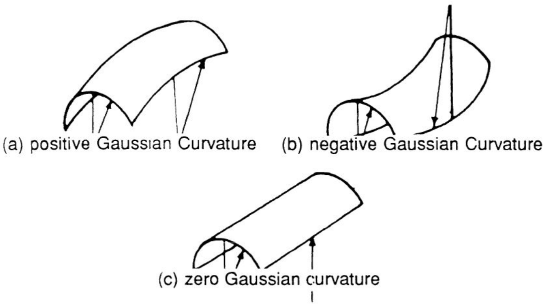
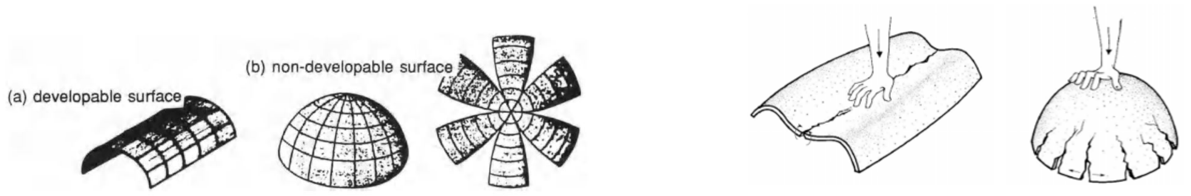

(a) positive Gaussian Curvature
(b) negative Gaussian Curvature
(c) zero Gaussian curvature

Figure 2.7: Gaussian curvature definition adapted from [6]

### 2.3.2 Developable shells

Another criteria, correlated with the shell stiffness is its developability property, which translates into the difficulty of turning the shell´s middle surface into a planar configuration by imposing an external force. Double curvature shells are, in general, non-developable, which is to say, a higher external energy is necessary for the structure to crumble into a planar configuration, as tension cracks are generated. In contrast, single curvature shells fall into a planar layout with only one crack due to the absence of significant in-plane stresses generated by the second curvature.



(a) Developable surfaces vs Non-Developable surfaces, adapted from [6]

(b) Collapse of non-developable and developable structures, adapted from [1]

Figure 2.8: Developabilty of shell structures

Hence, non-developable structures collapse due to tension, regarding concrete, while developable structures crumble due to bending. As it can be perceived, the second curvature adds extra bearing capacity to the shell.

### 2.3.3 Generated Surfaces

The way the shell´s surface is generated can also be considered to categorize shells. This classification can be valuable while designing and analysing shells, as surfaces from the same family share the common characteristics and behaviour, regarding construction and/or the way the loads are carried. Surfaces can be characterized as surfaces of revolution, surfaces of translation, ruled surfaces and free surfaces.

#### Surfaces of Revolution

The surface is generated by the revolution of a plane curve, denominated by meridional curve, about an axis, the axis of revolution. From this method, shapes such as cylinders, domes, semi-spheres can be generated.

#### Surfaces of Translation

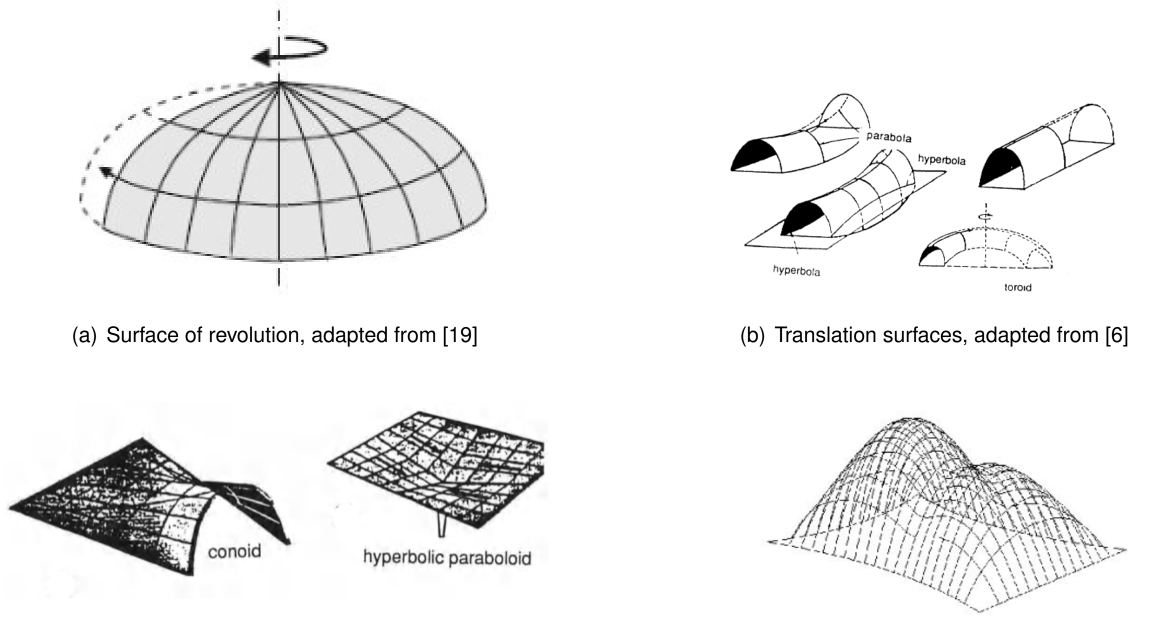
Surfaces of translation are generated by sliding a plane curve along another plane curve. The orientation of the former curve must remain constant. The plane curve which defines the translation mode of the 1st plane curve is denominated as generator of the surface. The generator of the surface can also be a straight line.

### Ruled Surfaces

Rules Surfaces are generated by sliding a straight line through a surface while touching 2 opposing borders, which can assume the form of a plane curve. Each straight line that defines the surfaces is not necessarily perpendicular or parallel to the plane curves that generate the surface. As the surface is formed through straight elements it is easier to be built, thus reducing costs associated with formwork and labor.

### Free Surfaces

Free surfaces do not follow any of the criteria above for its generation, as they are cannot be described by straightforward equations. Hence, they are usually generated through the use of NURBS (Non Uniform Rational B-spline), which can be applied to any geometry.



(a) Surface of revolution, adapted from [19]

(b) Translation surfaces, adapted from [6]

(c) Ruled Surfaces, adapted from [6]

(d) Free form surfaces, adapted from [19]

Figure 2.9: Classification of shell geometries regarding they generation process

## 2.4 Advantages and Disadvantages in modern engineering

Shells are an adequate example on how to provide resistance to a structure through its form without adding significant weight. Not only that, but while stresses in slabs´ sections develop linearly, resulting in zones that are not effectively solicited, shells develop uniform stresses throughout each section, behaving as membranes, if well designed. Therefore, if a geometrical configuration is adequately chosen, the material´s resistant capacity potential will be maximized. It was also observed by Heinz Isler that for the same amount of material, double curvature shells can withstand thirty times more weight than than a slab, without presenting significant deformation [20]. That is to say, shells are structurally more efficient than slabs.

On one hand, shells are adequate solutions to cover high spans due its structural efficiency which may suppress the need to design vertical elements, thus increasing the functionality of the built space. Obviously, this solution will be more adequate when applied, for instance, to massive roof-spans such as in industrial buildings, pavilions, sports centers, theaters, stadiums etc... [6]. Not only that, shells are more aesthetically pleasing, cause a positive impact on the viewer, provide an natural connection with the outside, through the entry of light and offer some acoustic advantages, and material versatility.

On the other hand, shell structures present significant disadvantages. Despite their structural efficiency, shell design can be an exhausting process, as previously mentioned. Experience and a deep understanding of shell structures is also required to effectively design them. Another reason that is linked with popularity of slabs over shells is the ease of construction [8]. Shell constructions require building experience, knowledge and a specific type of formwork. As each shell construction is singular, the costs associated with these types of construction are more expensive. Due to these reasons, it is normal that these structures are only built when applied to the examples mentioned above.

In conclusion, for large scale roof-covers, shell structures are, frequently, the best solution. Despite their high costs and slow construction, shells show considerable advantages considering the load carrying behavior efficiency, strength to weight ratio, degree of reserved strength and structural integrity, thickness to other dimensions ratio (span, radius of curvature), stiffness and containment of space [20].

# Chapter 3

# Overview of shell theory

So far, several conclusions were drawn regarding the behaviour of shells, their design, applications, limitations and challenges. In this chapter, a theoretical overview of shell theory will be presented focusing on uniformly loaded geometries, conventionally adopted in roof-covers.

## 3.1 Theoretical Overview

Before approaching each example, some background and notations regarding shell analysis generic to all examples must be presented. The geometrical features of shells, allow these elements to, not only resist out-of-plane loads through out-of-plane forces, but mostly through in-plane membrane forces. The scope of this project will focus on thin shells. As stated, the middle surface is the surface passing through the middle points of the thickness of the shell. If the thickness, t, of the shell is small when compared of the radius of curvature, R, then it is considered a thin shell. Although, there is no definitive rule to classify a shell as thin, one may consider it to be thin if the relationship $t/R$ is smaller than 1/200 in concrete or 1/300 in metallic structures [6].

### 3.1.1 Internal stresses in a Shell

For the sake of understanding the internal stress system in a shell, an infinitesimal element of an arbitrary shell is considered. Before prior analysis, a coordinate system must be chosen. As the middle-surfaces develops through two dimensions, a x,y,z coordinate system will be adopted, where the x-y-axis are perpendicular and belong to the middle surface of the shell (Gaussian coordinates). The infinitesimal element is obtained by cutting the shell with two pairs of principal plane curves, $ds_x$ and $ds_y$ apart, normal to the surface, that meet on the corners of the element, thus, perpendicular to each other. The principal plane curves have radii of curvature $r_x$ and $r_y$, respectively.

If the structure is subjected to external loading, temperature variations, deformations constraints, settlements or other actions, internal forces will develop. The shell stress field will be composed by two pairs of membrane stresses, two normal to the element´s face $\sigma_{xx}$ and $\sigma_{yy}$, and two tangent to the element´s face $\tau_{xy}$ and $\tau_{yx}$, and shear stresses, $\tau_{xz}$ and $\tau_{yz}$.

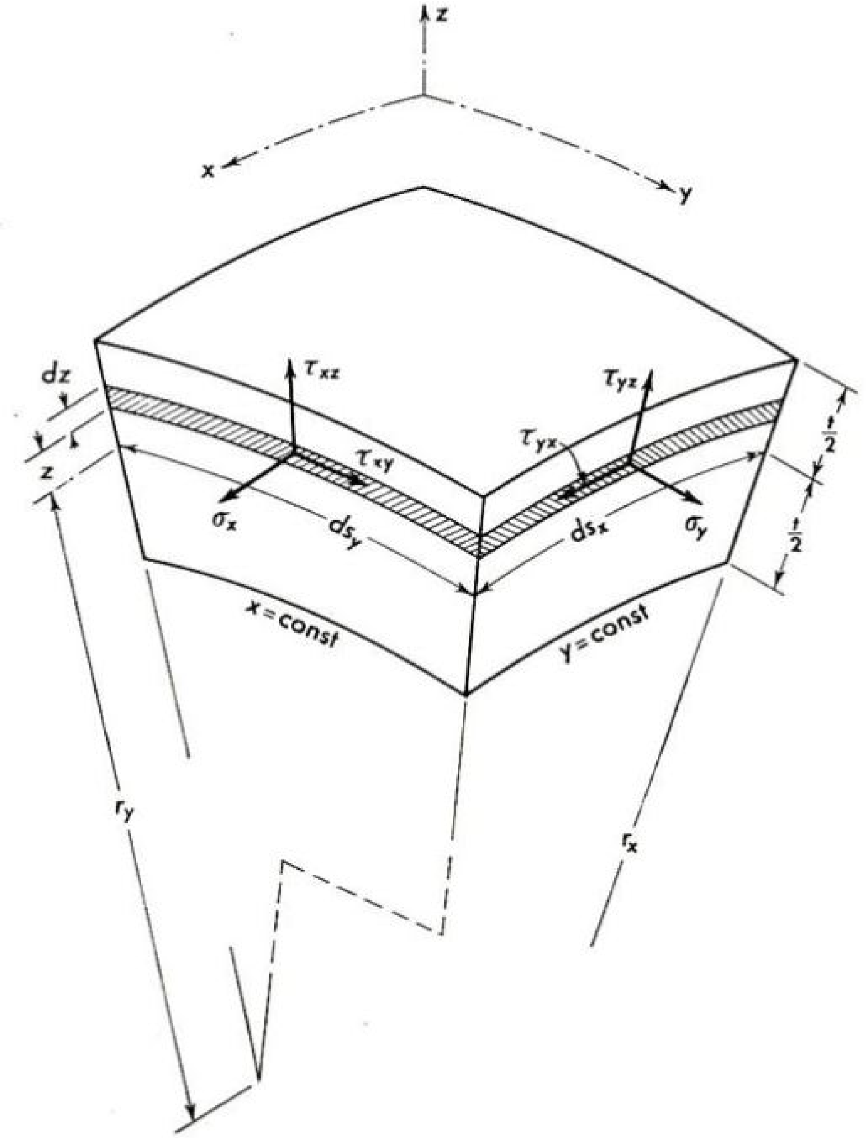


Figure 3.1: Internal stresses in an infinitesimal element of a generic shell [6]

The internal stresses in the infinitesimal element can be integrated over the shell´s thickness to obtain the shell´s governing equations. The distributed forces are statically equivalent to the stresses mentioned above, adapted from [8]. The terms $\frac{r_x+z}{r_x}$ and $\frac{r_y+z}{r_y}$ are necessary as the width of the borders changes throughout the shell´s thickness, due to its curvature.

For x = constant:

$$n_{xx} = \int_{-\frac{t}{2}}^{\frac{t}{2}} \sigma_{xx}(1+\frac{z}{r_y})\,dz \quad (3.1) \qquad n_{xy} = \int_{-\frac{t}{2}}^{\frac{t}{2}} \tau_{xy}(1+\frac{z}{r_y})\,dz \quad (3.2) \qquad v_{xz} = -\int_{-\frac{t}{2}}^{\frac{t}{2}} \tau_{xz}(1+\frac{z}{r_y})\,dz \quad (3.3)$$

$$m_{xx} = -\int_{-\frac{t}{2}}^{\frac{t}{2}} \sigma_{xx}\,z\,(1+\frac{z}{r_y})\,dz \quad (3.4) \qquad m_{xy} = -\int_{-\frac{t}{2}}^{\frac{t}{2}} \tau_{xy}\,z\,(1+\frac{z}{r_y})\,dz \quad (3.5)$$

for y = constant:

$$n_{yy} = \int_{-\frac{t}{2}}^{\frac{t}{2}} \sigma_{yy}\,(1 + \frac{z}{r_x})\,dz \quad (3.6) \qquad n_{yx} = \int_{-\frac{t}{2}}^{\frac{t}{2}} \tau_{yx}\,(1 + \frac{z}{r_x})\,dz \quad (3.7) \qquad v_{yz} = -\int_{-\frac{t}{2}}^{\frac{t}{2}} \sigma_{yz}\,(1 + \frac{z}{r_x})\,dz \quad (3.8)$$

$$m_{yy} = -\int_{-\frac{t}{2}}^{\frac{t}{2}} \sigma_{xz}\,z\,(1 + \frac{z}{r_x})dz \qquad (3.9) \qquad m_{yx} = -\int_{-\frac{t}{2}}^{\frac{t}{2}} \tau_{yx}\,z\,(1 + \frac{z}{r_x})\,dz \qquad (3.10)$$

Hence, the forces at stake can be divided into membrane forces and bending forces. Membrane forces act in the middle surface plane, are responsible for stretching and shrinking the shell, without producing additional bending moments. On the other hand, bending forces cause the bending and twisting of the shell´s cross section. The bending force field is composed by bending moments, twisting moments and shear forces.

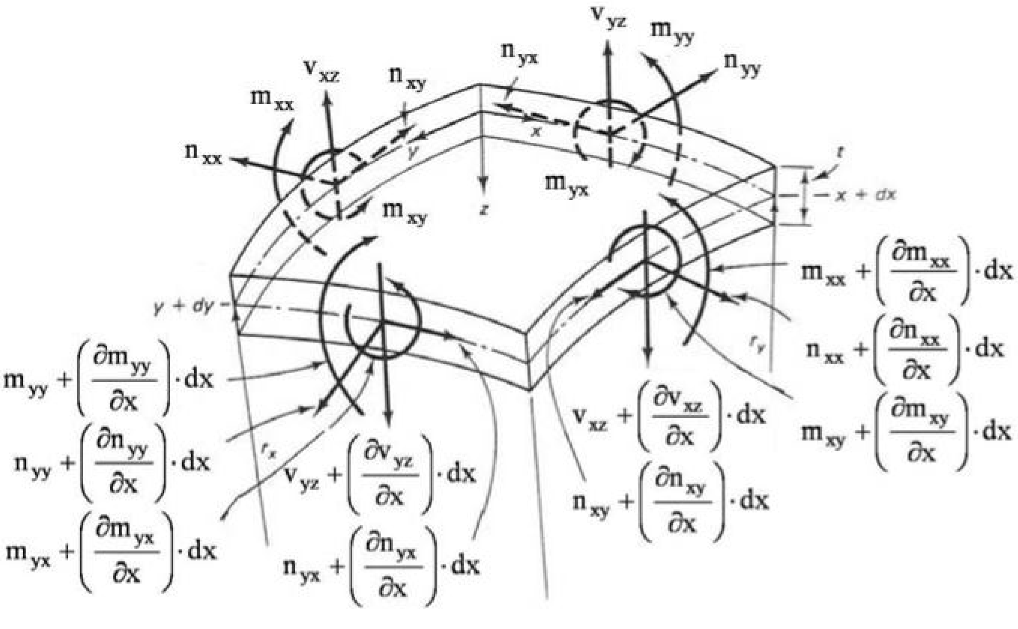

$v_{yz}$
$m_{yy}$
$n_{yx}$
$n_{yy}$
$v_{xz}$
$n_{xy}$
$m_{xx}$
$n_{xx}$
$x$
$y$
$z$
$m_{yx}$
$m_{xy}$
$t$
$x + dx$
$y + dy$
$r_y$
$r_x$
$m_{xx} + \left(\frac{\partial m_{xx}}{\partial x}\right) \cdot dx$
$n_{xx} + \left(\frac{\partial n_{xx}}{\partial x}\right) \cdot dx$
$m_{xy} + \left(\frac{\partial m_{xy}}{\partial x}\right) \cdot dx$
$v_{xz} + \left(\frac{\partial v_{xz}}{\partial x}\right) \cdot dx$
$n_{xy} + \left(\frac{\partial n_{xy}}{\partial x}\right) \cdot dx$
$m_{yy} + \left(\frac{\partial m_{yy}}{\partial x}\right) \cdot dx$
$n_{yy} + \left(\frac{\partial n_{yy}}{\partial x}\right) \cdot dx$
$v_{yz} + \left(\frac{\partial v_{yz}}{\partial x}\right) \cdot dx$
$m_{yx} + \left(\frac{\partial m_{yx}}{\partial x}\right) \cdot dx$
$n_{yx} + \left(\frac{\partial n_{yx}}{\partial x}\right) \cdot dx$

Figure 3.2: Internal forces in an infinitesimal element of a generic shell [21]

**Membrane Forces:** $n_{xx}$, $n_{yy}$, $n_{xy}$, $n_{yx}$

**Bending/Shear forces:** $m_{xx}$, $m_{yy}$, $m_{yx}$, $m_{xy}$, $v_{xz}$, $v_{yz}$

All things considered, an internally statically indeterminate problem arises, as there are six equilibrium equations for ten unknowns. The indeterminate problem can be solved by introducing the constitutive relationship of the elasticity theory. However, the solution would provide a system of differential equations of high order, which generally has no closed form solution.

### 3.1.2 Membrane Theory

The membrane theory of shells can be used to answer this problem effortlessly [6]. By admitting only membrane forces, the system of equations becomes determinate, as there are three equilibrium equations for only three unknowns, as $n_{xy} = n_{yx}$ due to symmetry. This significant simplification delivers fast and valuable guidelines that can be used at the beginning of the design stage or to compare the results with a desired behaviour. Nevertheless, although it may deliver satisfactory results on design stage, it may present some inconveniences regarding the incompatibility of the membrane´s solution with the shell´s boundary conditions, thus some corrections regarding bending theory have to be included.

To apply the membrane theory the following requirements have to be met [22]:

- The shell must be thin, according with the description previously presented. The thin shell´s flexibility will prevent the structure developing serious bending stresses, thus developing in-plane stresses to transmit external loads.
- The middle surface has to be continuous, to avoid bending and shear stresses from being developed.
- External loads are preferably uniform, as point loads cause localized bending in the structure. Point loads, normal to the surface, cannot be equilibrated by membrane forces, as the normal component of the membrane force tends to 0. Hence, shear and bending forces are introduced.
- The supports must be within the plane of the middle surface of the shell, to establish equilibrium. If the support can only equilibrate one of the components of the membrane force from the shell, elements can be added to withstand the load, see Figure 3.4.
- The boundary conditions have to be compatible with the displacements generated by the membrane behaviour in its boundary, which, generally, does not happen.

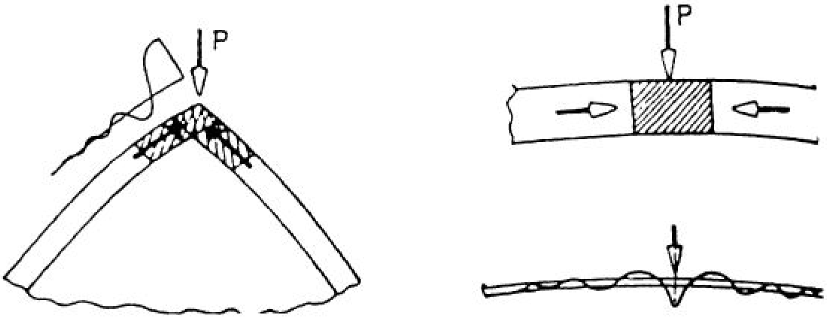


Figure 3.3: Bending introduced by discontinuities in the geometry and punctual loads in shell, adapted from [6]

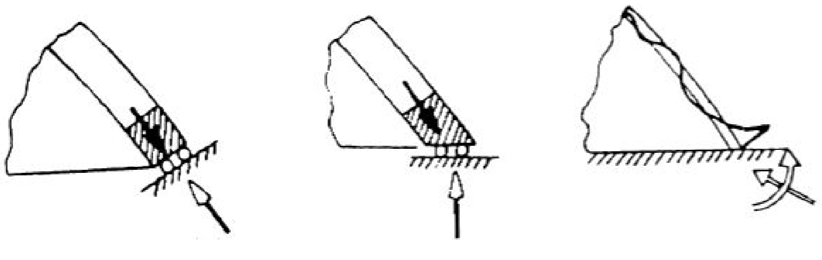


Figure 3.4: Bending introduced by angle between the shell and the support, adapted from [6]

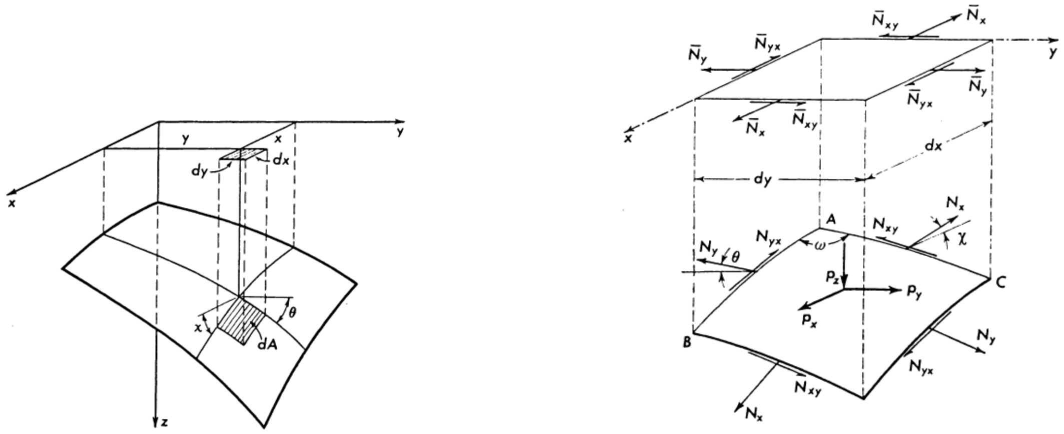
It is observed in Figure 3.3 that discontinuities introduce bending moments. However, bending rapidly fades, due to the flexibility of the shell; thus, bending is confined within the affected area.

As stated, the boundary conditions of the shell need to be respected by the results provided by the application of the membrane theory. Generally, bending does not play a significant role in shell design; however, when it is not negligible, the analysis of shell structures with the membrane theory needs to be corrected. As the results provided by the membrane theory will only be used on structures which behave almost as a perfect membrane, the study regarding the correction of the membrane theory will not be conducted.

## 3.2 Direct stresses on shells of arbitrary shape

Shells of revolution and translational shells such as cylinders, domes, hyperbolic and elliptic paraboloids, respectively, have well defined analytical solutions, regarding the stress and displacements calculation. Calculating the stresses of a structure with a prompt analytical solution provides a great insight about its structural behaviour. Therefore, it would be deeply relevant to find a prompt and simple analytical solution that could be applicable to certain geometries, namely, to geometries generated through the Bézier Surface formulation, as it will be done in Section 5.1. If this step is successful, the result could then be used to validate the Full Order Model and to assess the PGD application.The theory mentioned bellow was adopted from [8].

The middle surface of the shell can be described by a system of rectangular coordinates, where $Z$ is a function $z(x, y)$, see Figure 3.5(a). The geometry will be analytically defined through the application of Bézier Surfaces, which provide an analytical expression of z. The Bézier Surfaces concept will be better explained in the following chapters. Being a function, $x$ and $y$ can be also depicted as curvilinear coordinates on the shell.



(a) Shell of arbitrary geometry in rectilinear coordinates, adapted from [8]

(b) Shell element and forces projected on x and y plane, adapted from [8]

Figure 3.5: Shell projection in rectangular coordinates

The curvilinear lines, that define the element´s borders, are defined through the intersection of the shell´s middle surface with planes normal to the $x$ and $y$ axes. Hence, the originated element will not be rectangular, due to the curvature, and the membrane stresses will be described by a system of skew forces per unit of length, $N_{xx}$, $N_{yy}$, $N_{xy}$ and $N_{yx}$, with $N_{xy} = N_{xy}$, due to tensor´s symmetry as seen in Figure 3.5(b). The curvilinear lines meet at an angle $\omega$, which is defined as $\cos\omega = \sin\chi . \sin\theta$. It is important to remind that $N_{xx}$ corresponds to $n_{xx}$, used in Section 3.1 and so on. Furthermore, the relationship between the forces expressed in the two sets of coordinated systems is established:

For x constant:

$$N_{xx}\,\frac{dy}{\cos\theta}\,\cos\chi = \bar{N}_{xx}\,dy \qquad (3.11) \qquad\qquad N_{yx}\,\frac{dx}{\cos\chi}\,\cos\theta = \bar{N}_{yx}\,dx \qquad (3.12)$$

For y = constant:

$$N_{yy}\,\frac{dx}{\cos\chi}\,\cos\theta = \bar{N}_{y}\,dx \qquad (3.13) \qquad\qquad N_{xy}\,\frac{dy}{\cos\theta}\,\cos\chi = \bar{N}_{xy}\,dy \qquad (3.14)$$

The equilibrium of the element will be written with the new quantities, in the rectangular coordinate system, that is, the plane projections of the stress resultants, referred to the unit length of the projected line element $dx$ or $dy$. Not only the stresses, but the distributed loads per unit of area $p_x$, $p_y$, $p_z$ need to be written in terms of their plane projections $\bar{p_x}$, $\bar{p_y}$, $\bar{p_z}$. Without getting into great detail, the relationship between distributed loads is given by the ratio of the areas of a shell element $dA$ and of its projection $dx \times dy$, as following:

$$dA = \frac{dx}{\cos\chi}\,\frac{dy}{\cos\theta}\,\sin\omega = dx\;dy\;\frac{(1-\sin^2\chi\;\sin^2\theta)^{\frac{1}{2}}}{\cos\chi\;\cos\theta} \qquad (3.15)$$

therefore:

$$\frac{\bar{p_x}}{p_x} = \frac{\bar{p_y}}{p_y} = \frac{\bar{p_z}}{p_z} = \frac{dA}{dx\;dy} = \frac{(1-\sin^2\chi\;\sin^2\theta)^{\frac{1}{2}}}{\cos\chi\;\cos\theta} \qquad (3.16)$$

thus, the equilibrium equations, regarding plane projections are, respectively, in the x, y and z direction:

$$\frac{\partial\bar{N}_{xx}}{\partial x} + \frac{\partial\bar{N}_{xy}}{\partial y} + \bar{p}_x = 0 \qquad (3.17) \qquad\qquad \frac{\partial\bar{N}_{yy}}{\partial y} + \frac{\partial\bar{N}_{yx}}{\partial x} + \bar{p}_y = 0 \qquad (3.18)$$

$$\bar{N}_{xx}\frac{\partial^2 z}{\partial x^2} + 2\;\bar{N}_{xy}\frac{\partial^2 z}{\partial x\;\partial y} + N_{yy}\frac{\partial^2 z}{\partial y^2} = -\bar{p}_z + \bar{p}_x\frac{\partial z}{\partial x} + \bar{p}_y\frac{\partial z}{\partial y} \qquad (3.19)$$

There are two differential equations of the first order and one ordinary linear equation. To solve the problem, it may be advantageous to introduce an auxiliary variable to reduce the system to one second-order equation. The first two equations are identical with the conditions of equilibrium of a plane stress system, therefore the stress function method may be useful to answer this problem.

Three stresses are to be expressed as:

$$\bar{N}_{xx} = \frac{\partial^2 \Phi}{\partial y^2} - \int \bar{p}_x dx \qquad (3.20) \qquad\qquad \bar{N}_{yy} = \frac{\partial^2 \Phi}{\partial x^2} - \int \bar{p}_y dy \qquad (3.21)$$

$$\bar{N}_{xy} = -\frac{\partial^2 \Phi}{\partial x \partial y} \tag{3.22}$$

Assuming a compatible stress function that sets $N_{xx}$ and $N_{yy}$ to zero on $x = [0; L]$ and $y = [0; L]$, respectively:

$$\Phi(x, y) = kx(x - L)y(y - L) \rightarrow \Phi(x, y) = ku(u - 1)v(v - 1) \tag{3.23}$$

As $\bar{p}_y = \bar{p}_x = 0$, the equilibrium equations with the new variable are obtained:

$$\frac{\partial^2 \Phi}{\partial v^2}\, h_1(v, z_1, z_2) + \frac{\partial^2 \Phi}{\partial u^2}\, h_2(u, z_1, z_2) - \frac{\partial^2 \Phi}{\partial x \partial y}\, h_3(u, v, z_1, z_2) = -\bar{pz} \tag{3.24}$$

with

$$\frac{\partial^2 z}{\partial u^2} = h_1 = -v(B\,(6v - 5) - 2A\,(v - 1)) - B \tag{3.25}$$

$$\frac{\partial^2 z}{\partial v^2} = h_2 = -u(B\,(6u - 5) - 2A\,(u - 1)) - B \tag{3.26}$$

$$\frac{\partial^2 z}{\partial u \partial v} = h_3 = ((4A - 12B)\,u + 5B - 2A)\,v + (5B - 2A)\,u - 2B + A \tag{3.27}$$

$z(u, v)$ corresponds to the formulation regarding Bézier Surfaces with second degree polynomials, see Section 5.1, with $A = 4z_1$ and $B = 4z_2$ constants for each combination, $u = x/C$ and $v = y/C$, $C$ being a constant.

$$\frac{\partial^2 \Phi}{\partial v^2} = N_{xx} = ku(u - 1) \tag{3.28}$$

$$\frac{\partial^2 \Phi}{\partial u^2} = N_{yy} = kv(v - 1) \tag{3.29}$$

Finding the exact solution is not easy. The compatible solution for the given problem was sought by setting $N_{xx}$ and $N_{yy}$ to 0 on the borders perpendicular to their direction. However, it does not only reflects a relatively uniform loading, as one would expect for a shell with low height/span ratio, but also is constant along its perpendicular axis. The generated stresses were unable to follow the expected trend. Other trials, for other stress functions were run, but were not successful. The analytical solution would be a big advantage for the purpose of this work; however, finding the membrane stresses of an arbitrary thin shell using the membrane theory is out of the scope of this dissertation, and maybe impossible. The validation of the Full Order Model will partially be oriented by the exact membrane solution for generic elliptic paraboloids, whose geometry is defined by:

$$z(x,y) = \frac{x^2 c_1}{a^2} + \frac{y^2 c_2}{b^2} \tag{3.30}$$

whose stresses are given as, see [23]:

$$N_{xx} = -\frac{\bar{p}_z a^2 K}{c_1} C_x \tag{3.31}$$

$$N_{yy} = -\frac{\bar{p}_z b^2 K}{K c_2} C_y \tag{3.32}$$

$$N_{xy} = -\frac{\bar{p}_z ab}{\sqrt{c_1 c_2}} C_{xy} \tag{3.33}$$

with K being the coefficient:

$$K = \sqrt{\frac{1 + [(2c_1/a)(x/a)]^2}{1 + [(2c_2/b)(y/b)]^2}} \tag{3.34}$$

The geometrical parameters $a$, $b$, $c_1$ and $c_2$ correspond to half the span of the shell in the x and y direction, the height of shell´s center point, relatively to the border´s center point, and the border´s center point, respectively. The coefficients $C_x, C_y$ and $C_{xy}$ can be found in Appendix A.2.

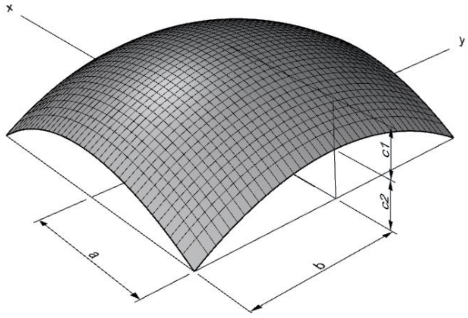


Figure 3.6: Eliptic paraboloid shell

# Chapter 4

# The PGD formulation

In the present stage of the project, the necessary background regarding shell characterization, behaviour behaviour and design, membrane theory and form-finding techniques in roof-cover applications has been presented. Thus, the methodology to find anti-funicular shapes for shell roofs is now be introduced.

The PGD is one of the approaches that allows complex systems to be simplified through the creation of Reduced Order Models, ROMs. The aim of these ROMs is to capture with significant accuracy the behaviour of the original model, the FOM, with significantly less computer costs. The problem at stake is to determine the parameters that minimize the bending energy of the shell, that is, the bending coefficient, $B_C$, a function from $\mathbb{R}^D$ to $\mathbb{R}$. The FOM solution is defined in its full multidimensional format, $\mathbf{B}_C$, with size $\prod_{i=1}^{D} n_i$, whose order corresponds to the number of parameters that define them. Whereas, the PGD solution, $\tilde{\mathbf{B}}_C$, is defined in its separated format, with size $M \times \sum_{i=1}^{D} n_i$.

To generate the models, $D$ parameters, $x_1, x_2...x_D$ are defined. Secondly, the sectional domains, that is, the spaces of each i-th parameter, $\Omega_i \subset \mathbb{R}$, with $n_i$ elements are discretized, with $i = 1, 2, ...D$. Thirdly, the mesh, for each parameter combination, is generated so the FEM can be carried out.

Assuming an arbitrary mesh with $nnodes$ nodes, to generate $\mathbf{B}_C \in \mathbb{R}^{n_1,\, n_2\, ...n_D}$, $B_C(x_1^{j_1}, x_2^{j_2}...x_D^{j_1})$ has to be calculated $\prod_{i=1}^{D} n_i$ times. As stated in Chapter 1, the resolution of this problem will scale exponentially with a higher number of parameters, $D$. Not only that, but for each iteration, a linear system of $dof = nnodes \times dofe$ equations, $u = K^{-1}\mathrm{f}$, needs to be assembled and solved. Being $dofe$ the number of degrees of freedom per node, $dof$, the total number degrees of freedom, $K$ the stiffness matrix, $u$ the nodal displacement vector and $\mathrm{f}$ the nodal force vector.

Furthermore, for a highly refined mesh, the time consumed on one iteration would turn the problem computationally impossible to solve. This would even more arduous if one´s intent was to also store $\mathbf{u}$ for each iteration, which would result in the storage of $\prod_{i=1}^{D} n_i \times dof$. The number of parameters used to describe the models will only go up to three, therefore the FOM solution is expected to be acceptable, despite the mentioned disadvantages.

Meanwhile, The PGD solves the problem at once and provides an answer, for any combination of parameters, in the parametric domain, building vademecums or digital abacuses, for which finding the parameters that best answer the problem have insignificant computational costs. For $\mathbf{x} = (x_1^{j_1}, x_2^{j_2}, ...x_D^{j_D}) \subset \Omega = \Omega_1 \times \Omega_2 ... \times \Omega_D \in \mathbb{R}^D$, the PGD calculates an approximation of $B_C$, $\tilde{B}_C$ as:

$$\tilde{B}_C(x_1^{j_1}, x_2^{j_2}, ...x_D^{j_D}) = \sum_{m=1}^{M} f_1^m(x_1^{j_i}) \times f_2^m(x_2^{j_2}) \times ... \times f_D^m(x_D^{j_D}) = \sum_{m=1}^{M} \prod_{i=1}^{D} f_i^m(x_i^{j_i}) \tag{4.1}$$

Where $m$ identifies each of the $M$ modes of the PGD solution, and $f_i^m$ a vector containing the value, $f_i^m(x_i^{j_i})$ for each parametric element, with $j_i = 1,\ 2\ ...\ n_i$. The number of elements to be stored in the separated tensor, $\tilde{\mathbf{B}}_\mathbf{C}$, becomes $M \times \sum_{i=1}^{D} n_i^m$, whose computational costs are insignificant when compared to the previous example, for a considerable $D$, thus avoiding the curse of dimensionality.

The approximate solution is built through a successive enrichment process without prior knowledge of the actual "correct" answer. Both the number of modes, $M$ and the functions $f_i^m(x_i)$ are unknown *a priori*. As $M$ increases, the approximate solution becomes more refined, increasing its resemblance with the exact solution. Thus, the error associated with it will most likely depend on the $M$ and the chosen sectional domains $\Omega_i$, see [2]. Additionally, the number of modes, $M$, generally required to acquire a reasonable solution does not depend on $D$, but it rather depends on the separability of the exact solution, and it is, generally, no more than a few dozens, see [24]. The separation is considered optimal when an exact solution can be represented with high accuracy by a reduced number of $M$. Otherwise, that is, when the solution is a strictly non-separable function, the PGD solver will carry the enrichment process until some some stopping criteria.

In the present case, $\tilde{\mathbf{B}}_C$, will be the result of several operations on separated tensors. The PGD methodology will be mainly applied to find separated and compressed representations of full multidimensional tensors, solve parametric linear system of equation and execute the Hadamard Division.

## 4.1 Progressive Construction of the Separated Representation

For an arbitrary step, $m \geq 1$, of the PGD method, the first $m - 1$ terms of the PGD approach have already been computed. Hence, the vectors in $\bigotimes_{i=1}^{D} f_i^m$, become the only unknowns of the problem, thus, a non-linear problems at each step. The strategy to solve this problem uses a sequential enrichment process based on the alternative direction strategy, to estimate the best $f_i^m$, for each iteration inside the mode, and on the greedy algorithm, whose role is to add new modes, sequentially computing the terms in the sum for $m = 1...M$, [3]. Being $\mathbf{T} \in \mathbb{R}^{n_1 \times n_2 \times ... \times n_D}$ a full multidimensional tensor, for each step, its separated approximation, $\tilde{\mathbf{T}} \in \mathbb{R}^{n_1 \times n_2 \times ... \times n_D}$ is given in its algebraic form by:

$$\tilde{\mathbf{T}}^m = \tilde{\mathbf{T}}^{m-1} + \bigotimes_{i=1}^{D} f_i^m \tag{4.2}$$

where:

$$\tilde{\mathbf{T}}^{m-1} = \sum_{m‘=1}^{m-1} \bigotimes_{i=1}^{D} f_\mathbf{i}^\mathbf{j} \tag{4.3}$$

## 4.2 Least-Squares PGD Separation and Compression

Despite following the same reasoning, the application of the Least-Squares PGD separation or the Least-Squares PGD compression will solely depend on the format of input and on some details on the algorithm: if the input, $\mathbf{T}$ is in full multidimensional format, the PGD separation is executed, whereas if the input is in a separated format, the PGD compression is carried out providing an output with less modes than the input, if successful. The main ideas of this topic will be concisely presented, for a deeper understanding of the notations and concepts applied, it is advised to read [3] and [5].

To perform these two types of approximations, the concept is the same: seek $\tilde{\mathbf{T}} \in V \subset \mathcal{L}_2(\Omega)$ so that the least squares projection of any $\mathbf{T} \in \mathcal{L}_2(\Omega)$ into $V$ is expressed as:

$$\tilde{\mathbf{T}} = \arg\min_{\mathbf{G}\in V} A(\mathbf{G}-\mathbf{T}, \mathbf{G}-\mathbf{T}) \tag{4.4}$$

Where the bilinear form $A(\cdot,\cdot)$, whose Hilbert space is characterized in [25], assumes values in $\mathcal{L}_2(\Omega) \times \mathcal{L}_2(\Omega)$ and for each sectional space assumes values in $\mathcal{L}_2(\Omega_i) \times \mathcal{L}_2(\Omega_i)$. For example, $A(\tilde{\mathbf{T}}, \tilde{\mathbf{T}}) = \sum_{m=1}^{M} \sum_{m=1}^{M} \bigotimes_{i=1}^{D} {f_i}^{m^T} \mathbf{A}_i f_i^m$. In this case, the Euclidean norm is performed in the algebraic tensorial format, thus, to characterize the bilinear form in each sectional space, $\mathbf{A_i} \in \mathbb{R}^{n_i \times n_i}$ becomes the identity matrix. Thereby, $\tilde{\mathbf{T}}$ needs to be found so that the quantity described in equation (4.5) is minimized. This is accomplished through the alternating direction strategy and the greedy algorithm.

$$A(\mathbf{T}-\tilde{\mathbf{T}}, \mathbf{T}-\tilde{\mathbf{T}}) = A(\tilde{\mathbf{T}}, \tilde{\mathbf{T}}) - 2A(\tilde{\mathbf{T}}, \mathbf{T}) - A(\mathbf{T}, \mathbf{T}) \tag{4.5}$$

### 4.2.1 Alternating Direction Strategy

For a random mode $m$, and iteration $p$, in order to solve the non-linear problem, that is, to find the functions $f_i^m$, an alternative direction strategy is considered. This iterative scheme consists in assuming that $f_{i\neq\gamma}^m$ are known, whereas an initial arbitrary value is assumed for $f_\gamma^{\mathbf{m,p}}$ is assumed to start the non-linear iterative process. The iterative process stops once a fixed point with a specified tolerance, $\eta$, see equation (4.6), or the maximum number of iterations is reached.

$$\eta > [({f_\gamma}^{m,p} - {f_\gamma}^{m,p+1})^T \mathbf{A}_\gamma (f_\gamma^{m,p} - f_\gamma^{m,p+1})]^{\frac{1}{2}} \tag{4.6}$$

Even if the application is to be carried out on multidimensional data in tensorial format, $T$, the reasoning is relatively the same. For $\tilde{\mathbf{T}}^m = \tilde{\mathbf{T}}^m = \bigotimes_{i\neq\gamma}^{D} f_i^m \otimes f_\gamma^m$ with $f_{i\neq\gamma}^m$ assumed to be known, the first component of equation (4.5), for a given $m$, is read as:

$$A(\tilde{\mathbf{T}}^{\mathbf{m}}, \tilde{\mathbf{T}}^{\mathbf{m}}) = (\prod_{i\neq\gamma}^{D} {f_i}^T \mathbf{A}_i f_i)\ ({f_\gamma}^{\intercal} \mathbf{A}_\gamma f_\gamma) = \alpha\ ({f_\gamma}^{\intercal} \mathbf{A}_\gamma f_\gamma) \tag{4.7}$$

Where $\alpha$ is constant for each iteration $p$. As for the second component in equation (4.5), both terms in $\mathbf{T}$ and $\mathbf{f}_{\mathbf{i}\neq\gamma}^{\mathbf{m}}$ are known and assumed to be known, respectively, it translates to:

$$A(\tilde{\mathbf{T}}^m, \mathbf{T}) = A(\bigotimes_{i \neq \gamma}^{D} f_i^m \otimes f_\gamma^{\mathbf{m}}, \mathbf{T}) = \mathbf{g}^T \mathbf{A}_\gamma f_\gamma \tag{4.8}$$

g is the bilinear form $A(\bigotimes_{i \neq \gamma}^{D} f_i^m, \mathbf{T})$, with $\mathbf{T}$ being a tensor of dimension $D$ and $\bigotimes_{i \neq \gamma}^{D} f_i^m$ a tensor of dimension $D-1$. Therefore, each component of $\mathbf{g}$, $\mathbf{g}_\gamma$ is defined as:

$$\mathbf{g}_\gamma = \mathbf{T}_{i=1 \ldots \gamma \ldots D}[\prod_{j \neq \gamma}^{D} \mathbf{A}_j f_j] \tag{4.9}$$

That is, each time the $\mathcal{L}_2$ product is calculated for a certain sectional domain $\Omega_{j \neq \gamma} \times \Omega_{j \neq \gamma}$, the tensor $\mathbf{T}$ $j^{th}$ dimension is contracted. This process occurs until $\mathbf{g}_\gamma$ a rank one tensor of dimension $n_\gamma$ is generated. Thus, equation (4.5), when minimized, is now expressed as:

$$(\alpha {f_\gamma}^\intercal \mathbf{A}_\gamma f_\gamma) = {\mathbf{g}_\gamma}^T \mathbf{A}_\gamma f_\gamma \tag{4.10}$$

Finally, for each iteration $p$, $f_\gamma^m$ is calculated as:

$$f_\gamma = \frac{1}{\alpha} \mathbf{g}_\gamma \tag{4.11}$$

### 4.2.2 PGD Modal Updating and Stopping Criterion for the Enrichment Process

After applying the alternating direction strategy successfully, a new mode $M$ is obtained and the approximation $\tilde{\mathbf{T}} = \sum_{m=1}^{M} \bigotimes_{i=1}^{D} f_i^m$ is known. To proceed with the calculation of a new mode, instead of finding an approximation of $\mathbf{T}$, the best approximation of the unsolved part of the problem, $\mathbf{T} - \tilde{\mathbf{T}}$, is sought. The stopping criterion for the enrichment process can be stopped if the approximation´s modes surpasses a user-defined threshold or if irrelevant modes start to be generated. The relevance of each mode is expressed by the ratio between its norm and, typically, the $1^{st}$ mode´s norm. The norm of a mode is calculated as:

$$\sigma^m = \prod_{i=1}^{D} \|{f_i}^m\| \tag{4.12}$$

Being $\|{f_i}^m\|$ the norm of each function, for a given mode, calculated as $({f_i}^{m^T} \mathbf{A}_i {f_i}^m)^{\frac{1}{2}}$. Thus, the enrichment process can be stopped when $\sigma^m < \eta^* \times \sigma^1$, where $\eta^*$ is a tolerance chosen by the user.

## 4.3 Hadamard Division

The Hadamard division consists in dividing two tensors, component by component. Although it is a simple task when the input is in its multidimensional format, it is not that straightforward if the tensors are in a separated fashion, [3]. The goal of performing such division with the PGD is to carry this operation without the need to rebuild the input in its multidimensional format. The resulting tensor,

$\tilde{\mathbf{F}} = [\bigotimes_{j\neq\gamma}^{D} f_j] \otimes f_\gamma$, obtained computing $1^{st}$ mode of the PGD Hadamard Division between two separate tensors, $\tilde{\mathbf{T}} = \sum_{m_T=1}^{M_T} \bigotimes_{i=1}^{D} t_i{}^{\mathbf{m_t}}$ and $\tilde{\mathbf{S}} = \sum_{m_S=1}^{M_S} \bigotimes_{i=1}^{D} t_i^{m_S}$ is denoted as:

$$\tilde{\mathbf{F}} = \tilde{\mathbf{T}} \oslash \tilde{\mathbf{S}} \tag{4.13}$$

or:

$$\tilde{\mathbf{T}} = \tilde{\mathbf{F}} \odot \tilde{\mathbf{S}} \tag{4.14}$$

If both sides are multiplied by a test separated tensor $\tilde{\mathbf{F}}^* = [\bigotimes_{j\neq\gamma}^{D} f_j] \otimes f_\gamma^*$, the only unknown to the problem will be $f_\gamma$. Thus, following a similar reasoning as in equation (4.5), in order to seek $\tilde{\mathbf{F}}$, the bilinear form in equation (4.15) should be null.

$$A(\tilde{\mathbf{F}} \odot \tilde{\mathbf{S}} - \tilde{\mathbf{T}}, \tilde{\mathbf{F}}^*) = 0 \tag{4.15}$$

which is the same as:

$$\sum_{m_s=1}^{M_s} \Big[ \prod_{j\neq\gamma}^{D} (f_j \odot s_j^{m_s})^T \mathbf{A}_j f_j \Big] (f_\gamma \odot s_\gamma^{m_s})^T \mathbf{A}_\gamma f_\gamma^* = \sum_{m_t=1}^{M_T} \Big[ \prod_{j\neq\gamma}^{D} (t_j^{m_t} \mathbf{A}_j f_j)^T \Big] (t_\gamma^{m_t}) \mathbf{A}_\gamma f_\gamma^* \tag{4.16}$$

in its simplified form:

$$\sum_{m_s=1}^{M_s} \alpha_{m_s} (f_\gamma \odot s_\gamma^{m_s})^T \mathbf{A}_\gamma f_\gamma^* = \sum_{m_t=1}^{M_T} \beta_{m_t} (t_\gamma^{m_t}) \mathbf{A}_\gamma f_\gamma^* \tag{4.17}$$

From this results:

$$f_\gamma = \sum_{m_t=1}^{M_T} \beta_{m_t} (t_\gamma^{m_t}) \oslash \sum_{m_s=1}^{M_s} \alpha_{m_s} s_\gamma^{m_s}) = g \oslash a \tag{4.18}$$

The stopping criteria for both the alternate direction scheme and the enrichment process follow the same principles mentioned in Section (4.2.1) and (4.2.2). To proceed to calculate a new mode, one should replace $\mathbf{T}$ for $\tilde{\mathbf{T}} - \tilde{\mathbf{F}}$. For a deeper understanding of this topic, it is advised to read references [3] and [5].

## 4.4 Parametric System of Linear Equations

The common approach to this problem would be to rebuild $\tilde{\mathbf{K}}$ and $\tilde{\mathbf{b}}$ into the multidimensional and then proceed to solve each equation as $\mathbf{u} = \mathbf{K}^{-1}\mathbf{b}$ for each parameter combination. Instead, the goal is to solve a parametric linear system of equations, only once, and deliver a solution for every parameter combination in a separated fashion, without the need to rebuild the intervening tensors in a multidimensional format. The parametric linear system of equations is described as:

$$\tilde{\mathbf{K}}\tilde{\mathbf{U}} = \tilde{\mathbf{B}} \tag{4.19}$$

The separated tensor $\tilde{\mathbf{K}} \in \mathbb{R}^{n \times n \times n_1 \times \ldots n_D}$ is composed by a non-parametric, $\sum_{m_K=1}^{M_k} K_0^{m_K}$, with $\mathbf{K}_0 \in \mathbb{R}^{n \times n}$, and parametric component, $\sum_{m_K=1}^{M_k} \bigotimes_{i=1}^{D} k_i^{m_K}$:

$$\tilde{\mathbf{K}} = \sum_{m_K=1}^{M_k} \mathbf{K_0} \bigotimes_{i=1}^{D} k_i^{m_K} \tag{4.20}$$

As the previous tensor, the separated tensor $\tilde{\mathbf{B}} \in \mathbb{R}^{n \times n_1 \times \ldots n_D}$ is composed by a non-parametric, $\sum_{m_b=1}^{M_b} b_0^{m_K}$, with $b_0 \in \mathbb{R}^n$, and parametric component, $\sum_{m_b=1}^{M_b} \bigotimes_{i=1}^{D} b_i^{m_b}$:

$$\tilde{\mathbf{U}} = \sum_{m_b=1}^{M_b} \mathbf{b_0} \bigotimes_{i=1}^{D} \mathbf{b}_i^{m_b} \tag{4.21}$$

The solution $\tilde{\mathbf{U}} \in \mathbb{R}^{n \times n_1 \times \ldots n_D}$ will follow the same structure. It will be composed by an non-parametric, $\sum_{m_u=1}^{M_u} u_0^{M_u}$, with $u_0 \in \mathbb{R}^n$, and parametric component, $\sum_{m_u=1}^{M_u} \bigotimes_{i=1}^{D} u_i^{m_u}$:

$$\tilde{\mathbf{U}} = \sum_{m_u=1}^{M_u} u_{\mathbf{0}} \bigotimes_{i=1}^{D} u_i^{m_u} \tag{4.22}$$

The solution will be built through successive enrichment, using the alternate direction scheme, providing two rank one tensor approximations, at a time for each mode. One related to the non-parametric part, and the other associated with the parametric part of $\tilde{\mathbf{U}}^{m_U} = u_{\mathbf{0}}^{\mathbf{m_U}} \bigotimes_{i=1}^{D} u_{\mathbf{i}}^{\mathbf{m_U}}$.

To calculate the parametric rank-one approximation, the procedure is the same as for the Hadamard Division. That is, for a given test function $\tilde{\mathbf{U}}^* = u_{\mathbf{0}} \otimes [\bigotimes_{j \neq \gamma}^{D} u_{\mathbf{j}}] \otimes u_\gamma^*$, the following bilinear form has to be null:

$$A(\tilde{\mathbf{K}}.\tilde{\mathbf{U}} - \tilde{\mathbf{B}}, \tilde{\mathbf{U}}^*) = 0 \tag{4.23}$$

which can be rearranged as:

$$A(\tilde{\mathbf{K}}.\tilde{\mathbf{U}}, \tilde{\mathbf{U}}^*) = A(\tilde{\mathbf{B}}, \tilde{\mathbf{U}}^*) \tag{4.24}$$

and, as in [3], developed to:

$$\begin{split} \sum_{m_K=1}^{M_K} [(K_{\mathbf{0}}{}^{m_k}.u_{\mathbf{0}})^T \mathbf{A_0} \mathbf{u_0}] \prod_{j \neq \gamma}^{D} \left[(k_j^{m_K} \odot u_j)^T \mathbf{A_j} u_{\mathbf{j}}\right] \left[(k_\gamma^{m_K} \odot u_\gamma)^T \mathbf{A}_\gamma u_\gamma^*\right] = \\ \sum_{m_B=1}^{M_B} \left[(b_{\mathbf{0}}{}^{m_B})^T \mathbf{A_0} u_{\mathbf{0}}\right] \prod_{j \neq \gamma}^{D} \left[(B_j^{m_b})^T \mathbf{A_j} u_{\mathbf{j}}\right] \left[(b_\gamma^{m_B})^T \mathbf{A}_\gamma u_\gamma^*\right] \end{split} \tag{4.25}$$

and by grouping the constants, the bilinear form is denoted as:

$$\sum_{m_K=1}^{M_K} \alpha_{m_K} \left[(k_\gamma^{m_K} \odot u_\gamma)^T \mathbf{A}_\gamma u_\gamma^*\right] = \sum_{m_B=1}^{M_B} \beta_{m_B} \left[(b_\gamma^{m_B})^T \mathbf{A}_\gamma u_\gamma^*\right] \tag{4.26}$$

implying that:

$$\Big[\sum_{m_K=1}^{M_K} \alpha_{m_K} k_\gamma^{m_K} \odot u_\gamma)\Big] = \Big[\sum_{m_B=1}^{M_B} \beta_{m_B} b_\gamma^{m_B}\Big] \tag{4.27}$$

resulting, for $\forall u_\gamma^*$:

$$u_\gamma = \Big[\sum_{m_B=1}^{M_B} \beta_{m_B} b_\gamma^{m_B}\Big] \oslash \Big[\sum_{m_K=1}^{M_K} \alpha_{m_K} k_\gamma^{m_K})\Big] = g \oslash a \tag{4.28}$$

To calculate the non-parametric rank-one approximation, the procedure is the same as for the Hadamard Division. That is, for a given test function $\tilde{\mathbf{U}}^* = u_\mathbf{0}^* \otimes [\bigotimes_{i=1}^{D} u_\mathbf{i}]$, the following bilinear in equation (4.23) form has to be null. From that it results:

$$\sum_{m_K=1}^{M_K} [(K_\mathbf{0}{}^{m_k}.u_\mathbf{0})^T \mathbf{A_0} u_\mathbf{0}{}^*] \prod_{i=1}^{D} \Big[(k_i^{m_K} \odot u_i)^T \mathbf{A_i} u_\mathbf{i}\Big] = \sum_{m_B=1}^{M_B} \Big[(b_\mathbf{0}{}^{m_B})^T \mathbf{A_0} u_\mathbf{0}\Big] \prod_{j=1}^{D} \Big[(b_i^{m_b})^T \mathbf{A_u} u_\mathbf{i}\Big] \tag{4.29}$$

that, through the same manipulation as before, becomes:

$$\Big[\sum_{m_K=1}^{M_K} \alpha_{m_K} K_\mathbf{0}{}^{m_k}.u_\mathbf{0}\Big]^T \mathbf{A_0} u_\mathbf{0}{}^* = \Big[\sum_{m_B=1}^{M_B} \beta_{m_B} b_\mathbf{0}{}^{m_B}\Big]^T \mathbf{A_0} u_\mathbf{0}^* \tag{4.30}$$

As both $\alpha_{m_K}$ and $\beta_{m_B}$ are scalars, the result is given, $\forall u_\mathbf{0}^*$, as the solution, $u_\mathbf{0}$, compatible with the equation below:

$$\Big[\sum_{m_K=1}^{M_K} \alpha_{m_K} K_\mathbf{0}{}^{m_k}\Big].u_\mathbf{0} = \sum_{m_B=1}^{M_B} \beta_{m_B} b_\mathbf{0}{}^{m_B} \tag{4.31}$$

Firstly, the space iteration (non-parametric) rank-one approximation is carried, followed by the parametric approximation. The stopping criteria for the alternate direction scheme is similar as the one on Hadamard Division. As for the stopping criteria, the reasoning continues to follow the same principle as the operations previously studied. As a new mode is sought, $\mathbf{B}$ have to be updated to $\mathbf{B}^m = \mathbf{B} - \mathbf{U}^{m-1}$.

# Chapter 5

# Implementation

## 5.1 Generation of the Geometry

The necessity of generating the shell´s geometry in a swift way is mandatory to promptly analyze the FOM´s results. There are several suitable approaches to complete this task. One may consider using Bézier Surfaces, B-Splines or NURBS. For a simple understanding of these three methods one can picture the Bézier curves as the grandparent of curve representation, the B-Splines as the parent and finally the NURBS as the children. Where each descendant carries the ability to assume the form of its ascendant/s. All of this methods allow the user to create $C^1$ and $C^2$ type curves.

Bézier Surfaces are a generalization of Bézier Curves. Bézier Curves allow the generation and the analytical definition of curves as a polynomial. Through the introduction of $p$ control points, a $n = p - 1$ degree polynomial is generated defining the curve. Therefore, Bézier Curves and Bézier Rational Curves present a basic approach to generate geometries such as, single and double curvature, quadratic and circular shells, which for the scope of this project is enough. Nevertheless, if free-form geometries are to be considered, the Bézier formulation would not be suited for this task. A higher degree of a curve implies a smoother curve, but also, higher computational costs. Not only that, but if a control point needs to be updated or removed the entire configuration of the curve is updated, which can be troublesome [26]. That is to say, the Bézier formulation only allows a global control of the curve. Finally, the curve starts and ends in the first and last control point, respectively. Bézier Surfaces allow the creation of non-developable, double curvature translational shell, that, in terms of membrane behaviour, are efficient, if well designed. For a more effective representation of complex free-form curves, one can consider B-Splines or NURBS.

All in all, by having a set of $p \times q$ = $(n + 1) \times (m + 1)$ control points $\mathbf{P}$, a surface $S$ composed by the translation of polynomials of degree $(m, n)$, in each direction, is expressed as following, through the Bernstein polynomials [27]:

$$z = S(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_i^n(u)\, B_j^m(v)\, P_{i,j} \tag{5.1}$$

$$B_i^n(u) = \binom{n}{i} u^i (1-u)^{n-i} \tag{5.2}$$

and,

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} \tag{5.3}$$

Equation (5.1) has the following matricial form:

$$\underset{1\times 1}{S(u,v)} = \underset{1\times p}{u} \times \underset{p\times q}{M_u} \times \underset{q\times p}{P} \times \underset{p\times q}{M_v} \times \underset{q\times 1}{v^{\mathsf{T}}} \tag{5.4}$$

with:

$$M_u(i,j) = M_v = \begin{cases} \binom{n}{i}\binom{n-1}{j}(-1)^{n+i-j} & 0 \le i \le n, \quad 0 \le j \le n-i \\ 0 \quad \text{otherwise} \end{cases} \tag{5.5}$$

and $[u]$ and $[v]$ translate into $[1 \ u^1 \ u^2 ... u^i ... u^n]$ and $[1 \ v^1 \ v^2 ... v^i ... v^m]$, respectively. The variables $u$ and $v$ translate the same information as $x, y$ but in a dimensionless fashion, that is, $u(x) = x/x_{max}$ and $v(y) = y/y_{max}$, assuming values, between 0 and 1.

Finally, the following properties of Bézier are enumerated, adapted from [26]:

- The convex hull property: the curve is contained in the convex hull formed by the control points $P_i$;
- transformation invariance: rotations, translations, and seatings are applied to the curve by applying them to the control points;
- endpoint interpolation: The starting and ending point of the curve match the first and last control point $C(O) = P_0$ and $C(I) = P_{n+1}$;
- Global control property: By changing one control point, the entire configuration of the curve changes;
- Continuity property: The degree of the curve´s polynomial, $n$ will solely depend on the number of control points, $n+1$, that always generates a continuously differentiable curve;

## 5.2 Hypotheses on the geometrical and mechanical properties of the shell

The generated surfaces will take into consideration the following assumptions regarding geometry, mechanical properties, boundary conditions, load conditions, analysis and parametrization::

- Essentially, only double curvature, translational, non developable shells will be considered as an attempt to minimize bending, *a priori*;
- The material following properties are considered to be uniform, $E = 31 \times 10^9 Pa, \nu = 0.20$, thickness, $t = 0.15m$, and mass density, $\rho = 2500 \ kg/m^3$;
- The shell´s plan view is a square with a span, $L = 10m$.
- Taking into consideration the height/span ratio from other shell works, the height of any point of the shell will be no more than $h = 3.5m$.

- The pinned corners were smoothed to control the stresses provided by the FEM and to follow the guidelines regarding the angle between the support and the shell, see Figure 3.4;
- The structure has 4 axis of symmetry;
- No openings will be considered
- Only the self-weight will be considered;
- The drilling degree of freedom will not be considered, as the height/span ratio is not significant;
- Five possible parameters associated with specific points of the shell can be considered: $z_1$ is associated with the height of the shell´s middle control point, $z_2$ is linked with the height referring to each of the border´s middle control point, $z_3$ is a parameter associated with the height of the control points located on the quarter of the shell´s diagonal, $z_4$ is associated with the height of the shell in its borders quarters´, $z_5$ is linked to the height of the control points between the position of $z_1$ and $z_2$. The height of the control points in those location are given by the functions $A$, $B$, $C$, $D$ and $E$, see Figure 5.1;

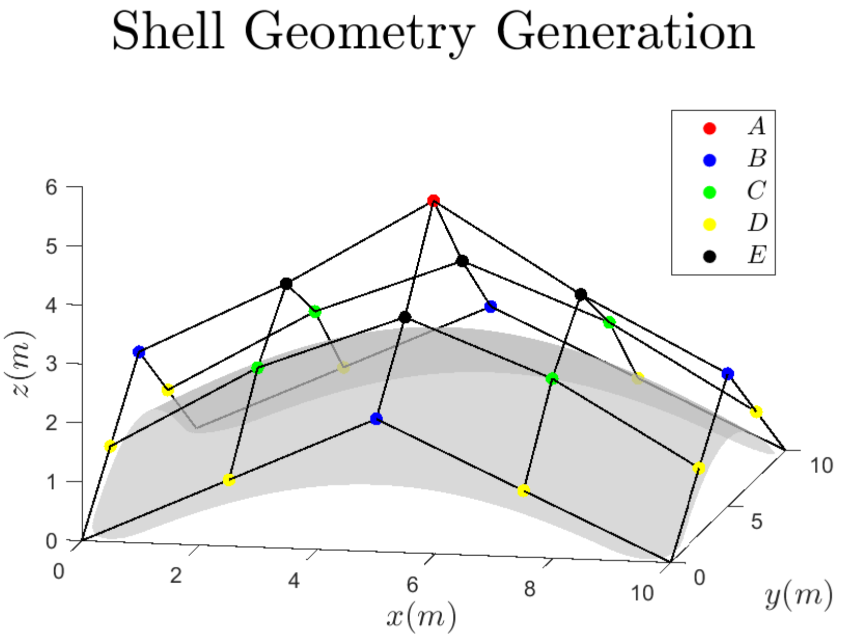


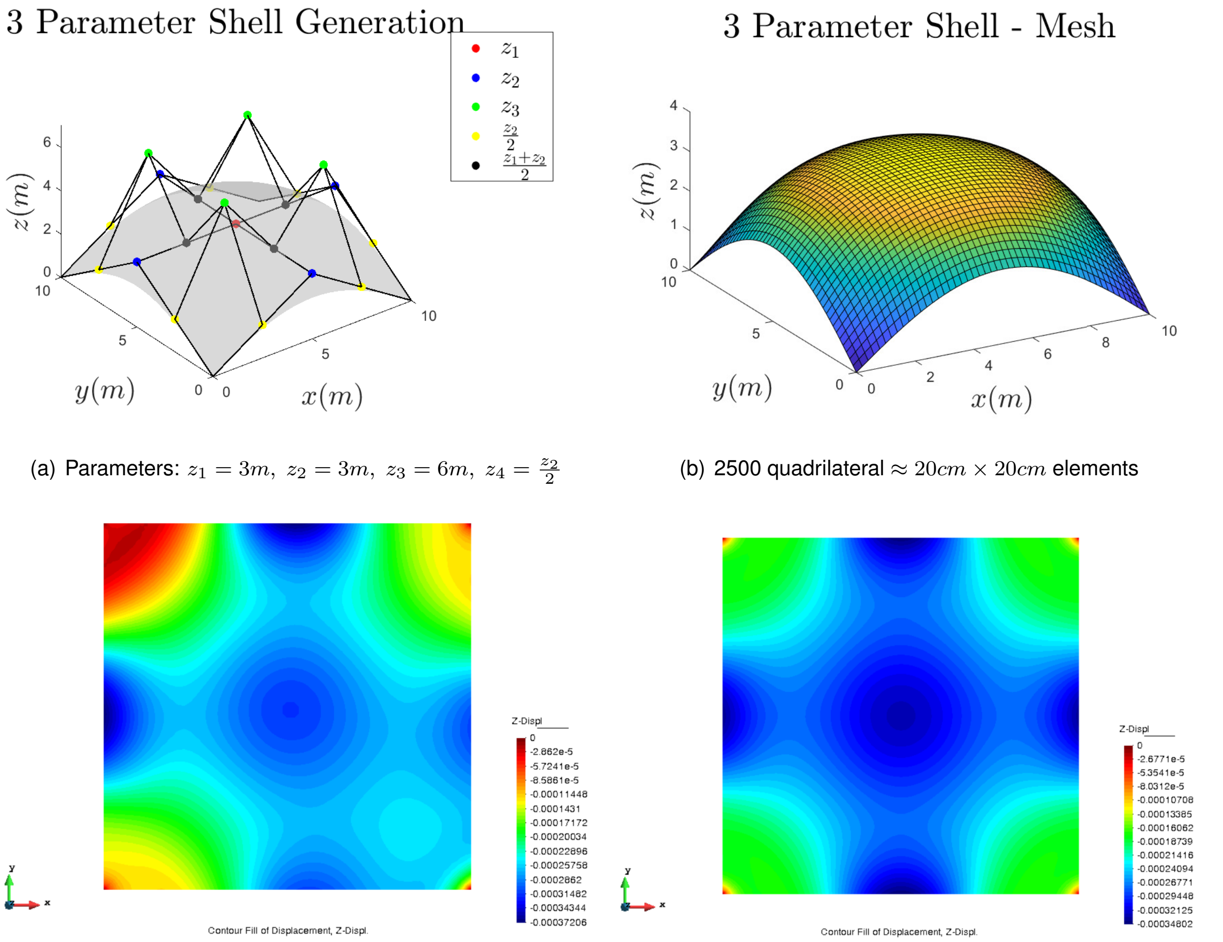
Figure 5.1: Arbitrary shell generated with 4 parameters, $A = 5,\ B = 2.25,\ C = 2.55,\ D = 1.25,\ E = 3.6$

## 5.3 Full Order Model

As the PGD will create a ROM of the Full Order Model, it is necessary the FOM´s obtained results. As several simulations must be carried out to obtain the results for different parameters, a prompt approach regarding geometry generation and analysis must be executed. The FOM geometry will be generated through the application of Bézier Surfaces. The $B_C$ will be obtained by FEM routines, originally provided by CIMNE, [4], regarding shell structures and following the Mindlin-Reissner formulation.

### 5.3.1 Generation of the mesh

By setting a Bézier Surface through an uniform distribution of the parameters $u$ and $v$ 4-node quadrilateral mesh can be generated directly through the Bézier formulation. However, for certain combinations of the parameters, once the height of the shell starts to be considerable, the results in terms of displacements become incompatible with the problem´s symmetry, specially near the pointy corners, see Figure 5.2. This problem can arise from the high concentration of stresses on the corners or the shear locking phenomena on meshes with $1^{st}$ approximations.



(a) Parameters: $z_1 = 3m,\ z_2 = 3m,\ z_3 = 6m,\ z_4 = \frac{z_2}{2}$

(b) 2500 quadrilateral $\approx 20cm \times 20cm$ elements

(c) Z - Displacements (m) - 4 Node Quadrilateral mesh (2500 elements)

(d) Z - Displacements (m) - 6 Node Triangular mesh (2740 elements)

Figure 5.2: Mesh with $1^{st}$ degree approximations (c) vs Mesh with $2^{nd}$ approximations (d)

A h-refinement around the corners and a p-refinement on the whole domain can be performed to solve the problem. As it can be seen in Figure 5.2, the problem was strictly related to the type of element used. The results provided with the quadrilateral mesh violate the symmetry of the problem, thus, the mesh was discarded. From the three available routines, two of them cause shear locking, namely the ones with $1^{st}$ degree approximations, hence the only solution was to use meshes with $2^{nd}$ approximations.

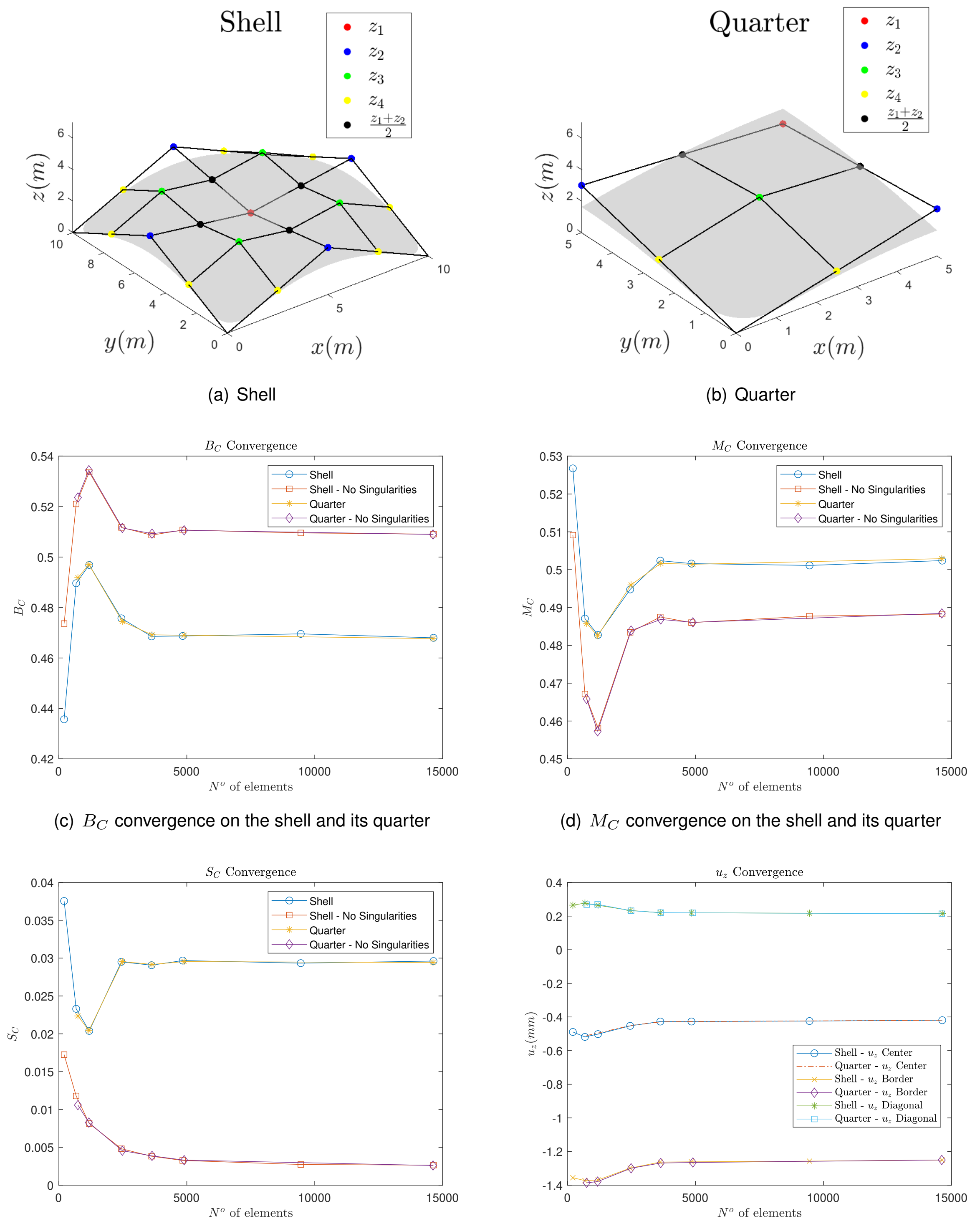
A flat non-uniform triangular $2^{nd}$ degree mesh, with higher discretization around the corners, will be generated on GMSH,[28]. As the maximum height of the shell will not be expressive, the mesh will be ”inflated”, through the Bézier Surface formulation with $S(u, v)$ on each node, to assume the desired shape. It would be convenient to take advantage of the structure´s symmetry to drastically decrease the time and storage spent in assembling and solving the stiffness matrix and executing the PGD operations. For that reason, only a quarter of the structure will be analyzed, $x, y \in [0; L/2]$. The simplification at stake can go as far as only analyzing an eighth of the structure, however, there would be a significant loss of generality towards the analysis of bi-symmetrical structures, and the computational gains are not remarkable. Restraining the study of the structure to its quarter, implies adding boundary conditions on the borders of the new structure, therefore, the following restraints were imposed. For $x = L/2$: $u_x = 0$ and $\theta_{xx} = 0$ as in [29] and for $y = L/2$: $u_y = 0$ and $\theta_{yy} = 0$ as in [29].

### 5.3.2 Validation of the Mesh

Validating the mesh corresponds to validate the Full Order Model. Validating the FOM translates in acquiring reasonable results that can be perceived as ”correct” and used as a reference. The CIMNE routine relies on the Direct Method to solve the fundamental equation. As the discretization of the mesh is more refined, the memory and time needed to assemble and solve the stiffness matrix increases drastically. Therefore, choosing a suitable mesh is a key aspect on building the FOM and apply the PGD. Therefore, a convergence study is conducted out between the whole structure and its quarter, for an increasing number of elements in the mesh, regarding $B_C$, $M_C$ (membrane coefficient) and $u$, see Figure 5.3.

Singularities can lead to significant errors on the calculation of the energy coefficients. Assuming that local disturbances remain local, the effect of the singularities was controlled, by not considering the stiffness of the elements around those areas, for energy calculation, see Figure 5.3. The results presented allow to safely assume that, at least, both models are answering the same question, thus, the simplification was successful. It is also noticeable the impact of not considering the corners when calculating the energy coefficients. To finally validate the model, the stresses obtained by application of the membrane theory presented on Section 3.2 were applied. By assuming $c_1 = c_2 = 1.5$, the parameters that would best shape a Bézier Surface into a elliptic paraboloid were estimated. For $A = 4$, $B = 2.6$ and $C = 3$ a similar geometry with sum of square, $SSE = 133m^2$, calculated along $101 \times 101$ surface points, was obtained. The average weight per unit of area, $\bar{p}_z$ was multiplied by a factor to take into consideration the curvilinear nature of the shell, by $\sum \mathrm{f}/(\rho \times t \times g \times L^2) = 1.1$. As both shells are symmetrical, by validating the stresses on the x direction, the y direction was also validated. In Figure 5.4 the stresses provided by the model are compared to the analytical ones for a similar structure.

For $y = L/2$, the small differences near the structure´s border can be explained by the introduction of $m_{xx}$ bending moments that arise near the borders. According to [23], no forces, normal to the edges, are allowed and all the loads carried along the free edges are by membrane shear forces down to the corner supports, and seen for both cases. For $x = L/2$ the obtained results through the FEM provide similar results with the analytical approach, starting from $y = 1.5m$. Prior to this distance, the FEM provides significantly inferior results when compared to the analytical method, being 0 at the border. The only difference between the two models resides on the boundary conditions, while the analytical model foresees the existence of stiff ”arches” on the borders, the FEM model operates with free borders. Thus,



(a) Shell

(b) Quarter

(c) $B_C$ convergence on the shell and its quarter

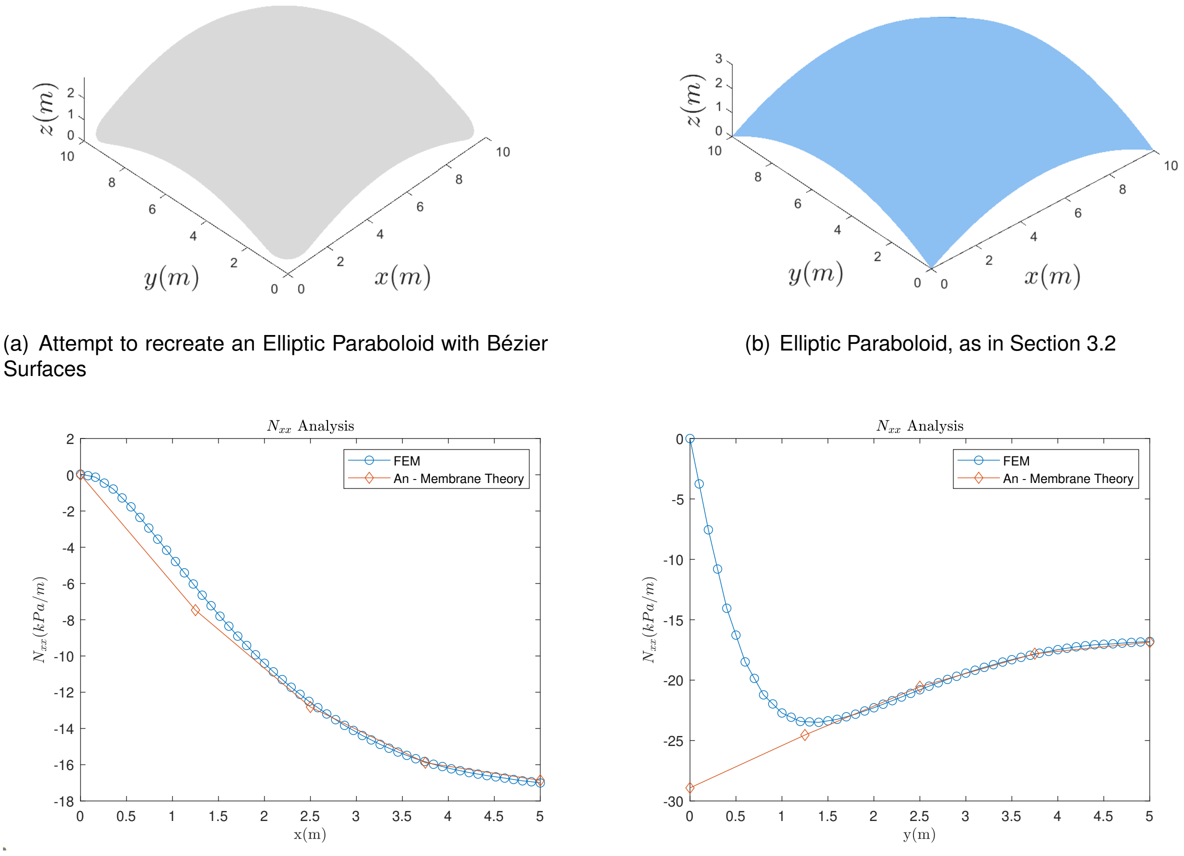
(d) $M_C$ convergence on the shell and its quarter

(e) $S_C$ convergence on the shell and its quarter

(f) $u_z$ convergence on the shell and its quarter

Figure 5.3: Convergence study for a shell and its quarter with parameters $z_1 = 2$, $z_2 = 3$, $z_3 = 3$ and $z_4 = 1.5$

the difference in results is explained by the existence of a reaction on the analytical model. All in all, the results lead to the conclusion that a $2^{nd}$ degree triangular mesh with 1913 nodes, with 5460 / 4 = 910 (quarter simplification) elements will provide reliable results, regarding the analytical approach. The mesh used to discretized the real problem the real problem is displayed in Figure 5.5.



(a) Attempt to recreate an Elliptic Paraboloid with Bézier Surfaces

(b) Elliptic Paraboloid, as in Section 3.2

(c) $N_{xx}$ for $y = L/2$

(d) $N_{xx}$ for $x = L/2$

Figure 5.4: Stress comparison between the model and membrane theory

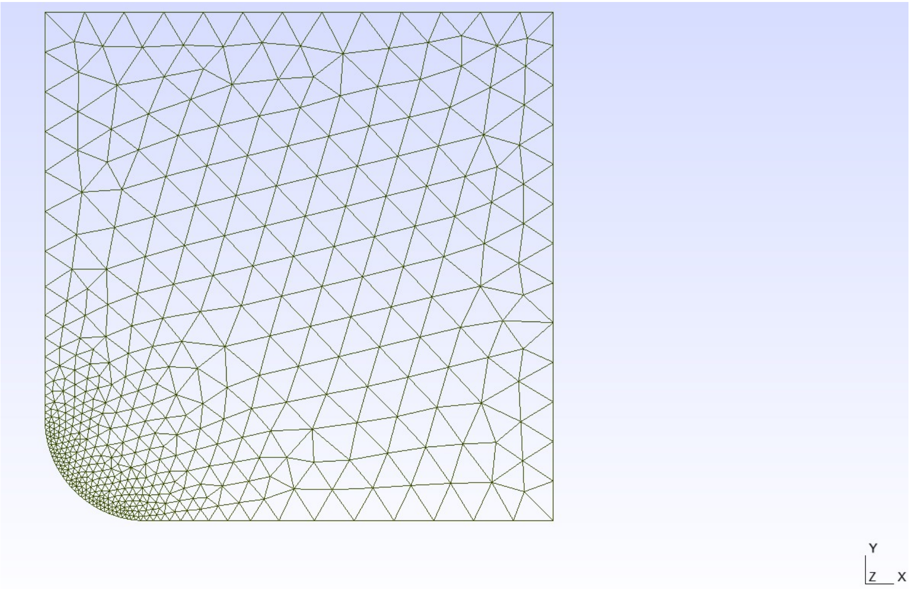


Figure 5.5: Mesh used to discretize the problem - $2^{nd}$ order triangular mesh (910 elements)

## 5.4 Variables at stake and Procedure

Several variables have been mentioned, but, not defined. In this section all the relevant variables will be formally defined. The variables of interest, that illustrate the behaviour of the structure in terms of bending, membrane and shear are the bending coefficient, $B_C$, the membrane coefficient, $M_C$ and shear coefficient $S_C$, expressed as:

$$B_C = \frac{\mathcal{U}_b}{\mathcal{U}} \quad (5.6) \qquad M_C = \frac{\mathcal{U}_m}{\mathcal{U}} \quad (5.7) \qquad S_C = \frac{\mathcal{U}_s}{\mathcal{U}} \quad (5.8)$$

where $\mathcal{U}_b$, $\mathcal{U}_m$, $\mathcal{U}_s$ and $\mathcal{U}$ are the bending strain energy, membrane strain energy, shear strain energy and total strain energy, considering the finite element method compatible formulation, which are defined as, for a given compatible displacement field $u$:

$$\mathcal{U}_b = \frac{1}{2}\int_\Omega (\epsilon_b(u))^T \; D_b \; \epsilon_b(u) \; d\Omega \quad (5.9) \qquad \mathcal{U}_m = \frac{1}{2}\int_\Omega (\epsilon_m(u))^T \; D_m \; \epsilon_m(u) \; d\Omega \quad (5.10)$$

$$\mathcal{U}_s = \frac{1}{2}\int_\Omega (\epsilon_s(u))^T \; D_s \; \epsilon_s(u) \; d\Omega \quad (5.11) \qquad \mathcal{U} = \mathcal{U}_b + \mathcal{U}_m + \mathcal{U}_s \quad (5.12)$$

with the material property operator $D_{j=b,m,s}$ for a isotropic material is defined as:

$$D_b = \frac{Et^3}{12(1-\nu^2)}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{(1-\nu)}{2} \end{bmatrix} \quad (5.13) \qquad D_m = \frac{Et}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{(1-\nu)}{2} \end{bmatrix} \quad (5.14)$$

$$D_s = \frac{5}{6}\frac{tE}{2(1+\nu)}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{5.15}$$

Considering $u_h$ an approximated displacement field expressed as:

$$u_h = Uu \tag{5.16}$$

with $U$ corresponding to the approximating displacement functions and $u$ the nodal displacement vector, one can express the strain $\epsilon$ as:

$$\epsilon = (\mathsf{D}U)u = Bu \tag{5.17}$$

where D is a differential operator. Thus, replacing equation (5.17) in equations (3.4), (5.7), (5.8), the following result in its generalization is obtained:

$$\mathcal{U} = \frac{1}{2}u^T \int_\Omega B^T DB \; d\Omega \; u \tag{5.18}$$

Taking into account that the stiffness matrix, $K$, of an element is defined as:

$$K_e = \int_{\Omega_e} B^T \; D \; B d\Omega_e \tag{5.19}$$

The equations of strain energy are as follows:

$$\mathcal{U}_b = \frac{1}{2}(u^T \; K_b \; u) \tag{5.20}$$

$$\mathcal{U}_m = \frac{1}{2}(u^T \; K_m \; u) \tag{5.21}$$

$$\mathcal{U}_s = \frac{1}{2}(u^T \; K_s \; u) \tag{5.22}$$

$$\mathcal{U} = \frac{1}{2}(u^T \; K \; u) \tag{5.23}$$

The nodal displacement vector is obtained by solving the fundamental equation (5.24), where $\mathrm{f}$ is the nodal force vector.

$$u = K^{-1}\mathrm{f} \tag{5.24}$$

To obtain $\mathbf{B_C}$, equation 5.24 has to be solved for each parameter combination. Whereas to obtain its separated counterpart $\tilde{\mathbf{B}}_\mathbf{C}$, the equation only has to be solved once as in (5.25), in a separated and tensorial fashion. Resulting in a separated tensor $\mathbf{u}$ with size $M(\mathbf{u}) \times (\sum_i^D n_i + dof)$, instead of $\prod_i^D n_i \times dof$:

$$\tilde{\mathbf{u}} = \tilde{\mathbf{K}}^{-1}\tilde{\mathbf{f}} \tag{5.25}$$

The multidimensional format stiffness and force tensors, $\mathbf{K_b},\; \mathbf{K_m},\; \mathbf{K_s}$ and $\mathbf{f}$ are assembled and compacted, as detailed in Section 5.4.1, in a custom made routine PGD_input. These tensors are separated in the PGD routine PGDTensorSeparation.m, resulting in $\tilde{\mathbf{K}}_\mathbf{b},\; \tilde{\mathbf{K}}_\mathbf{m},\; \tilde{\mathbf{K}}_\mathbf{s}$ and $\tilde{\mathbf{f}}$. To solve the fundamental equation through the PGD routine pgdLinearSolve.m, the separated tensors need to be rearranged between their non-parametric component and parametric components as in equations (5.26) and (5.27).

$$\tilde{\mathbf{K}}_\mathbf{j} = \sum_{m=1}^{M} K_0^m \prod_{i=1}^{D} \bigotimes f_i^m \tag{5.26}$$

$$\tilde{\mathbf{f}} = \sum_{m'=1}^{M'} \mathrm{f}_0^{m'} \prod_{i=1}^{D} \bigotimes f'^{m'}_i \tag{5.27}$$

$K_0^m$ and $\mathrm{f}_0^{m'}$ the non-parametric components with size $(dof \times dof)$ and $(1 \times dof)$, respectively, of each mode. And $f_i^m$ and $f'^{m'}_i$ the rank one tensors with size $n_i$ that constitute the parametric component of the stiffness and force tensors, respectively, for each mode. In order to save storage, the FOM will not produce the Full Order tensor $\mathbf{u}$, however, for some combinations, $u$ will be calculated to assess the error of the solution obtained from the pgdLinearSolve.m. This error can be measured according to equation (5.28), to estimate the number of modes, $m$, needed to deliver an accurate solution and to evaluate the effectiveness of this new PGD application. This error measure will be useful to characterize the quality of each PGD operation.

$$\epsilon^M(x_1^{j_1}, x_2^{j_2} ... \, x_D^{j_D}) = \frac{\|u^{FOM}(x_1^{j_1}, x_2^{j_2} ... \, x_D^{j_D}) - \sum_{m=1}^{M} \tilde{u}^m(x_1^{j_1}, x_2^{j_2} ... \, x_D^{j_D})\|}{\|u^{FOM}(x_1^{j_1}, x_2^{j_2} ... \, x_D^{j_D})\|} \tag{5.28}$$

Then, the energies are directly obtained by adapting equations (5.20), (5.21), (5.22), (5.23), to their separated tensor counterpart:

$$\tilde{\mathcal{U}}_j = \frac{1}{2}\tilde{\mathbf{u}}^T \tilde{\mathbf{K}}_\mathbf{j} \tilde{\mathbf{u}} \tag{5.29}$$

The product on equation (5.29) corresponds to the Hadamard product, that is, a component by component product that results in a tensor with $M(\tilde{\mathbf{u}}) \times M(\tilde{\mathbf{K}}) \times M(\tilde{\mathbf{u}})$ modes. Many of the modes of the resulting separated tensor do not offer relevant information, due to their small relative importance. Therefore, a truncation of the modes with relative importance inferior than a user-defined threshold is performed, before executing tensor compression with pgdCompression.m. Finally, $\tilde{\mathbf{B}}_C$ is calculated by performing the Hadamard Division, equation (5.30), a component by component division. Performing the Hadamard Division between two Full Order Model tensors is trivial in Matlab, however, if both are in a separated format, the pgdHadamardDivision.m routine can be executed.

$$\tilde{\mathbf{B}}_C = \tilde{\mathcal{U}}_B \oslash \tilde{\mathcal{U}} \tag{5.30}$$

The parameter combination that leads to a structure whose bending behaviour is minimized is identified through the pgdEvaluateSpaceIx.m routine, which essentially computes explicitly, at minimal computational costs, $\tilde{B}_C$. The parameters, $x$ that minimize the bending coefficient are expressed as:

$$x = (x_1^{j_1}, x_2^{j_2} ... x_D^{j_D}) = \arg\min_{x \in \Omega} \tilde{B}_C \tag{5.31}$$

Additionally, the thresholds used in each PGD operation are compiled in Table 5.1.

Table 5.1: Stopping criteria for each PGD operation

| Tolerance | pgdTensorSeparation | pgdLinearSolve | pgdCompression | pgdHadamardDivision |
|---|---|---|---|---|
| $\eta^*$ | $10^{-6}$ | $10^{-4}$ | $10^{-3}$ | $10^{-3}$ |
| $\max(M)$ | 100 | 100 | 100 | 100 |

### 5.4.1 Optimization and extension of routines

Two programs derive from the FEM routine provided by CIMNE, one creates the FOM, that delivers tensors with energy coefficients for a set of structures,$\mathbf{B_C}$, $\mathbf{M_C}$ and $\mathbf{S_C}$ and the other generates and folds the stiffness matrices into the PGD input. At its original form, the goal of the CIMNE program is to read the geometry of a given structure, perform the FEM, deliver the results in terms of displacements and stresses and create a GID, [30], post-processing file.

Firstly, the mentioned routine was only prepared to read customized files generated by GID, that is, if analyzing different structures was needed, several geometries may had to be manually and individually modelled on GID, to create an input for this routine. Therefore, the routine was altered in order to receive the desired parameters, receive the flat mesh provided by GMSH and ”inflate” the mesh to create the geometry according to the Bézier formulation. As the material properties, fixed nodes and flat mesh were constant, they were read only once on the main script. From that stage, the FEM routine would proceed to generate and solve automatically several structures. However, as the original routine was not conceived for this purpose, an optimization had to be carried out. The routine was not designed to solve a large number of structures, hence, it was consuming a great amount of time time to compute the stiffness matrices for each combination, due to sparse indexing, on the assemblage step. After time-optimization, a structure with 10 000 node mesh would be assembled and solve, with the additional computation of the energy coefficients in $1s$, instead of $20s$. A small program was also elaborated to read the mesh file created with GMSH, to obtain the nodes´coordinates and topology. The extension process of the routine is illustrated on Figure 5.6.

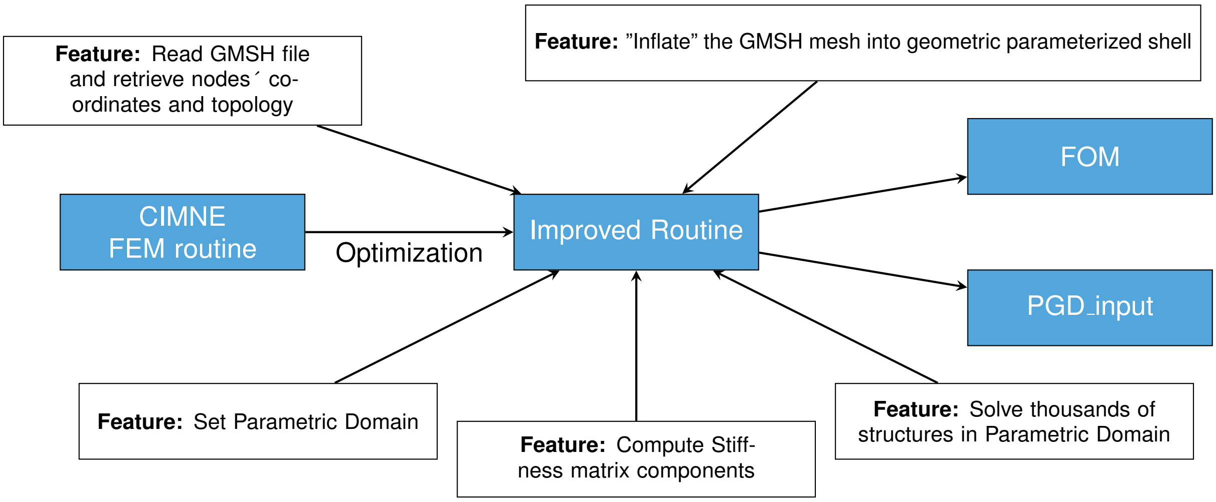

**Feature:** Read GMSH file and retrieve nodes´ coordinates and topology
**Feature:** ”Inflate” the GMSH mesh into geometric parameterized shell
FOM
CIMNE FEM routine
Optimization
Improved Routine
PGD_input
**Feature:** Set Parametric Domain
**Feature:** Compute Stiffness matrix components
**Feature:** Solve thousands of structures in Parametric Domain

Figure 5.6: Optimization and extension of FEM Matlab routine

From this step, to create the FOM, only the computation of the stiffness matrices, energy components and energy coefficients were left to be integrated. The FOM played an important role to choose an adequate mesh and boundary conditions, that lead to converging results. The flow chart associated with the FOM can be seen in Figure 5.7.

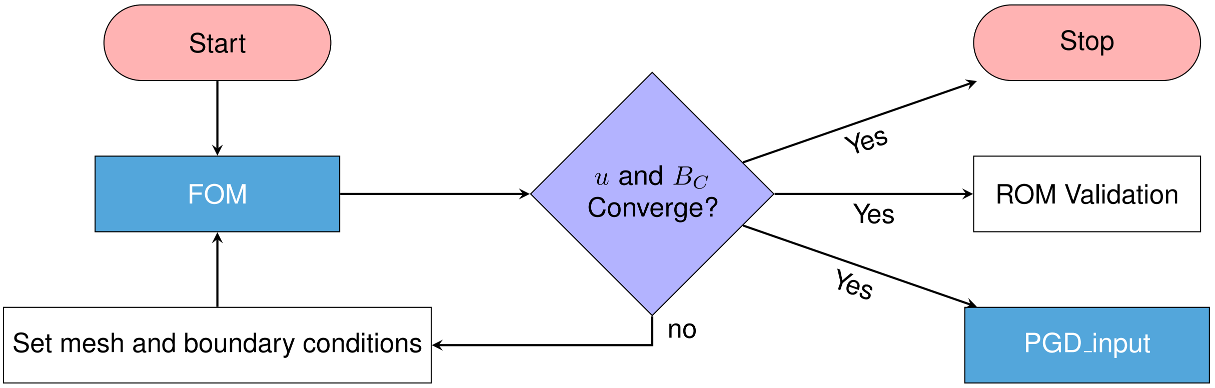


Figure 5.7: Mesh and boundary conditions retrieval algorithm

To create the PGD input, the computation of the stiffness matrices was also implemented. Due to their highly sparse nature, an optimization regarding memory usage was included. The stiffness matrices were turned into upper triangular matrices, UPM, then, only the non-zero elements were transferred to the stiffness tensors as rank-one tensors. However, depending on the combination, some positions that were previously occupied by non-zero elements could become zero, depending on the geometry of the configuration. This was a problem, because a constant non-parametric dimension of the stiffness tensor was assumed. To address the problem, due to the negligible number of ”oscillating zeros”, only the values located on the positions, whose indexes correspond to non-zero values of the $1^{st}$ iteration, were retrieved. It is crucial to highlight that this method is not bullet-proof, however, as it will observed, in Chapter 6 it does not lead to trackable errors. After the separation of the tensors, their rank-one non-parametric components had to be rearranged into matrices, with the indexes stored from the previous step, to serve as an input to the PGDLinearSolver. The flow chart associated with PGD_input can be seen in Figure 5.8:

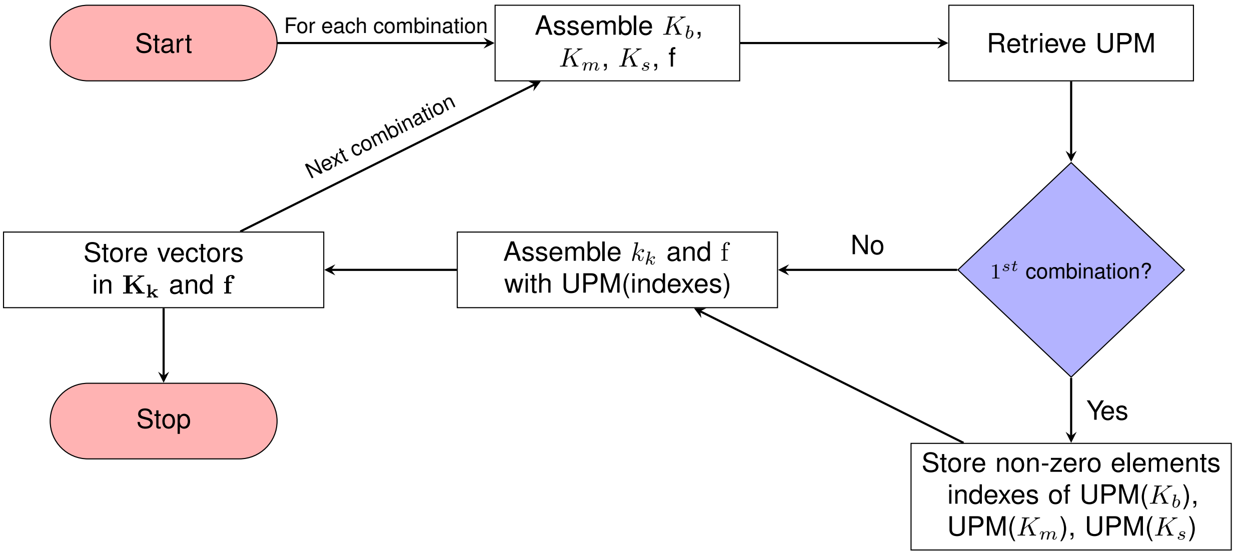

Start
For each combination
Assemble $K_b$, $K_m$, $K_s$, f
Retrieve UPM
$1^{st}$ combination?
No
Yes
Store non-zero elements indexes of UPM($K_b$), UPM($K_m$), UPM($K_s$)
Assemble $k_k$ and f with UPM(indexes)
Store vectors in $\mathbf{K_k}$ and $\mathbf{f}$
Next combination
Stop

Figure 5.8: PGD_input generating algorithm

All of the steps describing the implementation of the PGD can be better understood in Figure 5.9.

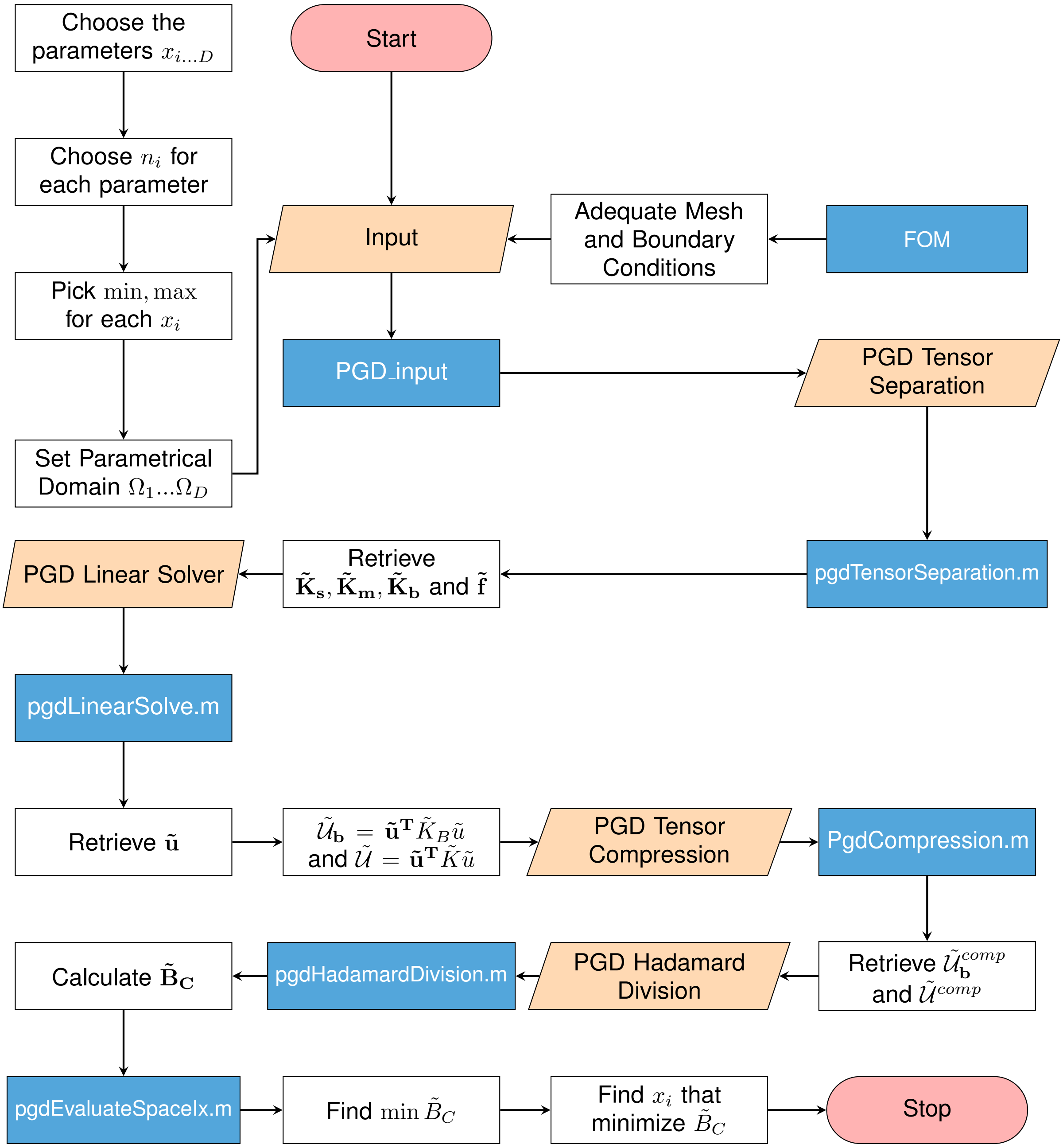

Choose the parameters $x_{i...D}$
Start
Choose $n_i$ for each parameter
Input
Adequate Mesh and Boundary Conditions
FOM
Pick $\min, \max$ for each $x_i$
PGD_input
PGD Tensor Separation
Set Parametrical Domain $\Omega_1 ... \Omega_D$
PGD Linear Solver
Retrieve $\tilde{\mathbf{K}}_\mathbf{s}, \tilde{\mathbf{K}}_\mathbf{m}, \tilde{\mathbf{K}}_\mathbf{b}$ and $\tilde{\mathbf{f}}$
pgdTensorSeparation.m
pgdLinearSolve.m
Retrieve $\tilde{\mathbf{u}}$
$\tilde{\mathcal{U}}_\mathbf{b} = \tilde{\mathbf{u}}^\mathbf{T} \tilde{K}_B \tilde{u}$ and $\tilde{\mathcal{U}} = \tilde{\mathbf{u}}^\mathbf{T} \tilde{K} \tilde{u}$
PGD Tensor Compression
PgdCompression.m
Calculate $\tilde{\mathbf{B}}_\mathbf{C}$
pgdHadamardDivision.m
PGD Hadamard Division
Retrieve $\tilde{\mathcal{U}}_\mathbf{b}^{comp}$ and $\tilde{\mathcal{U}}^{comp}$
pgdEvaluateSpaceIx.m
Find $\min \tilde{B}_C$
Find $x_i$ that minimize $\tilde{B}_C$
Stop

Figure 5.9: ROM´s generation algorithm

# Chapter 6

# Results and Discussion

## 6.1 Shell defined by 1 parameter

It is possible to promptly generate a panoply of different shells, while respecting their classification, with only one parameter, $z_1$, and without compromising the Bézier Surface formulation. By following the assumptions stated in Section 5.2 and by setting the majority of the control points as a function of $z_1$, such as:

$$S(u,v) = \begin{bmatrix} u^4 & u^3 & u^2 & u & 1 \end{bmatrix} \times \begin{bmatrix} 1 & -4 & 6 & -4 & 1 \\ -4 & -12 & -12 & 4 & 0 \\ 6 & -12 & 6 & 0 & 0 \\ -4 & 42 & 6 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \times \mathbf{P}(z) \times \mathbf{M} \times \begin{bmatrix} v^4 \\ v^3 \\ v^2 \\ v \\ 1 \end{bmatrix} \tag{6.1}$$

where the $z$ components of the control points, $\mathbf{P}(z_1)$, are:

$$\begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} & P_{04} \\ P_{10} & P_{11} & P_{12} & P_{13} & P_{14} \\ P_{20} & P_{21} & P_{22} & P_{23} & P_{24} \\ P_{30} & P_{31} & P_{32} & P_{33} & P_{34} \\ P_{40} & P_{41} & P_{42} & P_{43} & P_{44} \end{bmatrix} = \begin{bmatrix} 0 & D & B & D & 0 \\ D & C & E & C & D \\ B & E & A & E & B \\ D & C & E & C & D \\ 0 & D & B & D & 0 \end{bmatrix} \tag{6.2}$$

with $A = z_1$, $B = 5\sin(A.\frac{\pi}{2}.\frac{1}{8})$, $C = -\frac{A}{2} + 5\sin(A.\frac{\pi}{2}.\frac{1}{8})$, $D = \frac{B}{2}$, $E = 0.7A$. The generation of this surface is illustrated in Figure 6.1.

The parametric domain was set along 60 elements on $\Omega_1 \in [2; 7.5]$. The PGD approximation was straightforwardly carried on, as there is only 1 parameter and the application is relatively easy. The most important step of the approach, the separation of the stiffness and force tensors, was accomplished with few modes, as it can be seen on Appendix A.3. The results regarding the energy coefficient approximations and evolution of the shell´s area are displayed in Figure 6.2. In Figure 6.3, some geometries generated with this formulation are presented.

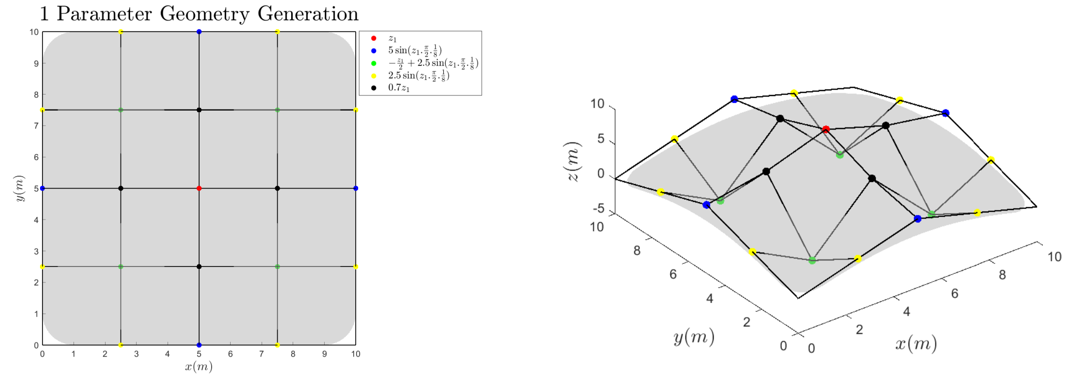
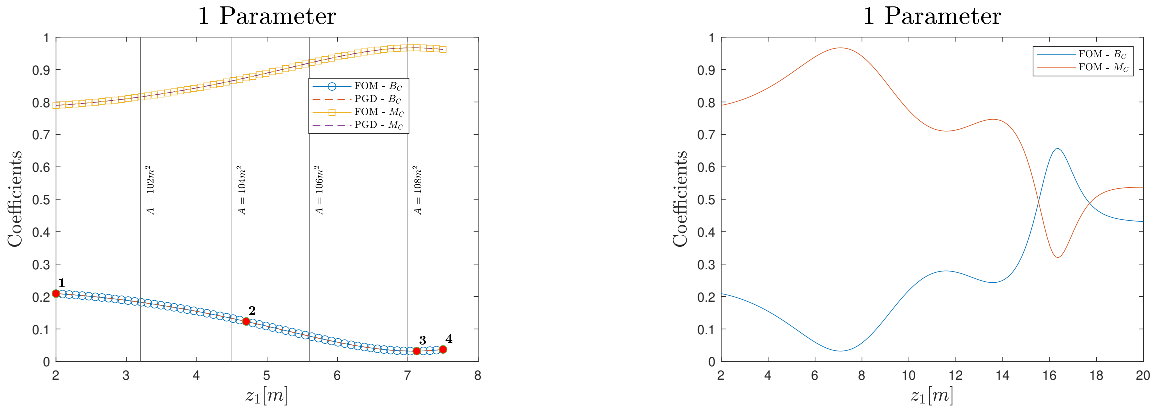

1 Parameter Geometry Generation
$z_1$
$5\sin(z_1.\frac{\pi}{2}.\frac{1}{8})$
$-\frac{z_1}{2}+2.5\sin(z_1.\frac{\pi}{2}.\frac{1}{8})$
$2.5\sin(z_1.\frac{\pi}{2}.\frac{1}{8})$
$0.7z_1$
$x(m)$
$y(m)$
$z(m)$

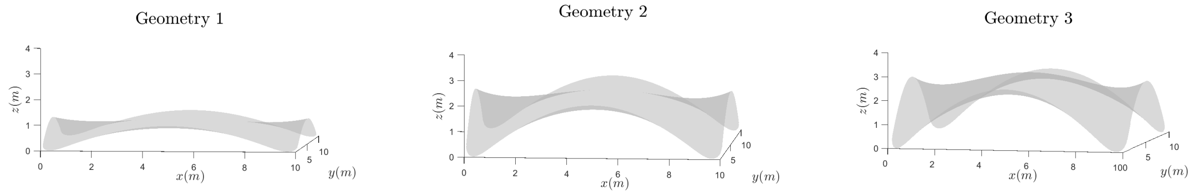
Figure 6.1: Arbitrary shell generated by 1 parameter with current formulation



(a) FOM and PGD approximation (1 mode)

(b) $z_1 \in [2;20]$

Figure 6.2: FOM solution and PGD approximation



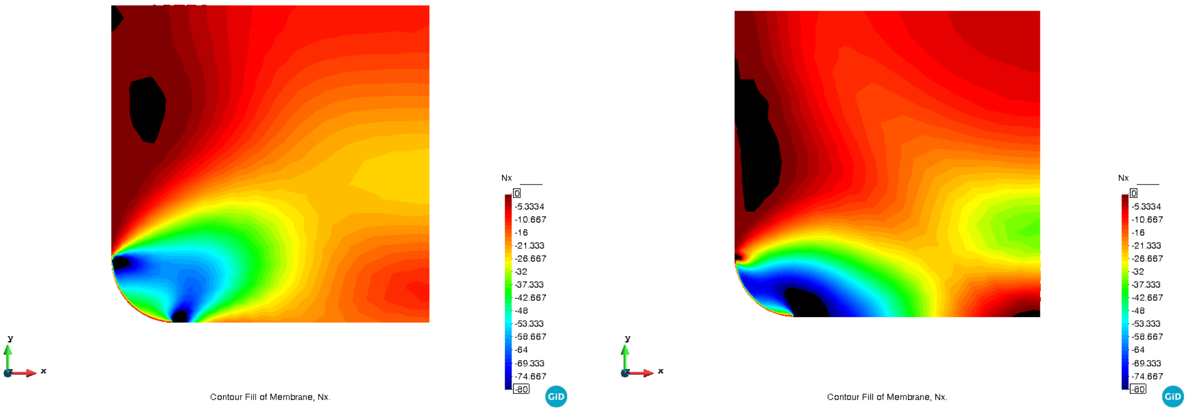
(a) $z_1 = 2$, $B_c = 0.2$

(b) $z_1 = 4.7$, $B_c = 0.15$

(c) $z_1 = 7.1$, $B_c = 0.03$

Figure 6.3: Geometries generated by with parameter

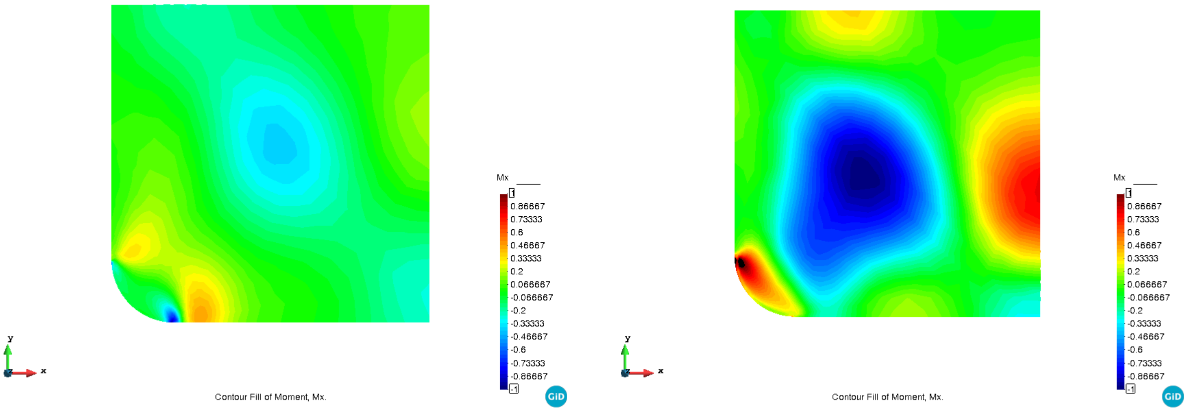
The implementation was successful, the ROM seems to match the results of the FOM with only 1 mode. The ROMs stresses and moments generated also match the FOM results, as seen in Figure 6.4, 6.5 and 6.6. This was expected, as there was only 1 parameter at stake. For that reason, the results provided will be used to test, once again, and strengthen the knowledge of the model. Starting from $z_1 = 7.5$, the geometrical configurations start to be physically inconceivable. Nevertheless, it is noticeable that bending behaviour reaches its lowest within the established domain. Through the configuration set on equation (6.2), the geometries tend to be flatter, for a lower $z_1$, thus having higher $B_C$. As polynomials of degree 4 are used, increasing the height of $z_1$ may not stimulate the membrane behaviour.



(a) $N_{xx} - (kN/m) : z_1 = 7.1m$

(b) $N_{xx} - (kN/m) : z_1 = 10m$

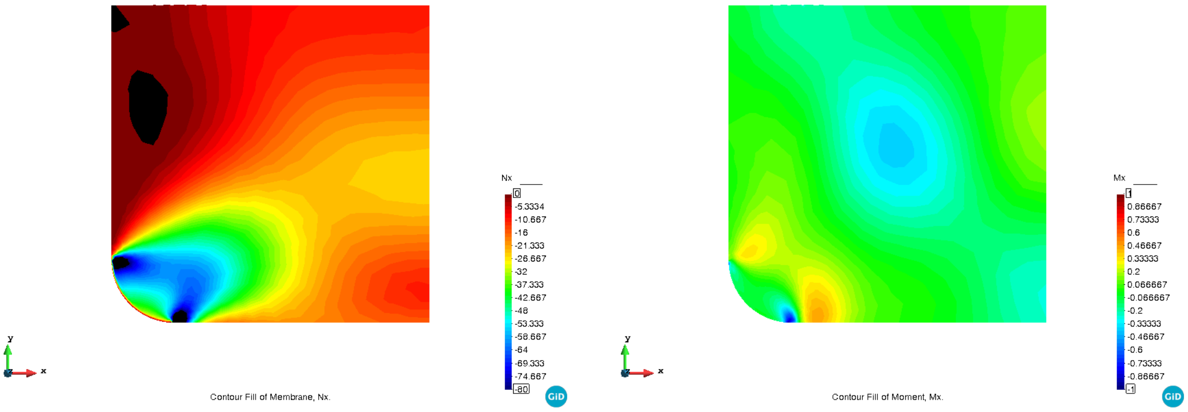
Figure 6.4: Membrane behaviour of the shell - 1 Parameter



(a) $m_{xx} - (kNm/m) : z_1 = 7.1m$

(b) $m_{xx} - (kNm/m) : z_1 = 10m$

Figure 6.5: Bending behaviour of the shell - 1 Parameter



(a) $\tilde{N}_{xx} - (kN/m) : z_1 = 7.1m$

(b) $\tilde{m}_{xx} - (kNm/m) : z_1 = 7.1m$

Figure 6.6: PGD stress and moment approximation

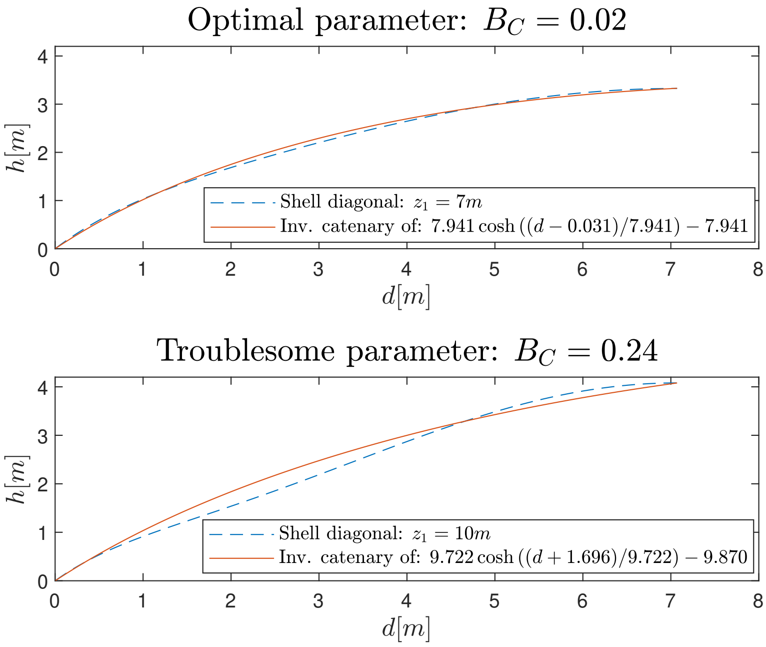


Figure 6.7: Shell´s diagonal and its catenary

For this type of formulation, the diagonal of the shell plays a significant role to carry the load to the supports, as the internal stresses converge there, Figure 6.3. Not only that, but it was observed in post-processing, that as $z_1 > 7.1$, a concentration of moments increases around the diagonal´s zone. Whereas, for $z_1 = 7.1m$ the moments are almost null in this area, Figure 6.5. This phenomenon can be be explained by the theory presented in Section 2.2.1. The inverted catenary is in full compression, by reducing the diagonal of the shell to a simple arch, if its configuration accurately resembles the shape of its corresponding inverted catenary, then, bending will be minimal. Otherwise, bending will increase. As seen in Figure 6.7, the optimal parameter´s diagonal configuration almost matches the configuration of its respective catenary arch. For a higher $z_1$, the shape of the diagonal starts to drift away from the catenary configuration, allowing for bending to increase.

The catenaries were calculated from a generalization of equation (2.1), in this case, for uneven supported arches, whose inputs were the supports´ Cartesian coordinates and the arches´ length that were calculated numerically. Note: for simplification the representation does not attend the corner simplification. From the stress analysis it is perceptible that minimal positive tension arises near the borders. Although the objective is to find antifunicular shells, this solution is reasonable, as the conditions on Section 5.2 imposed to circumvent this issue are not bulletproof. For this reason, the same study will be carried for 2 parameters,mainly, because of three aspects: 1) Erradicate traction 2)Reduce $B_C$ 3)Find new shapes.

## 6.2 Shell defined by 2 parameters - simple approach

For a simple implementation of the Bézier Surface formulation, one would only need to perform a translation of second degree polynomials, that is, only nine control points are necessary to define an arbitrary geometry. The shell´s geometry is described by two parameters, $z_1$ and $z_2$, as seen in Figure 6.8.

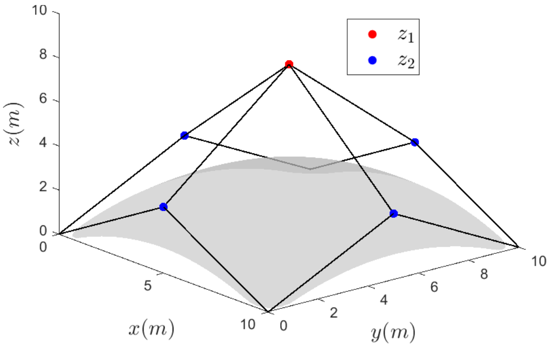


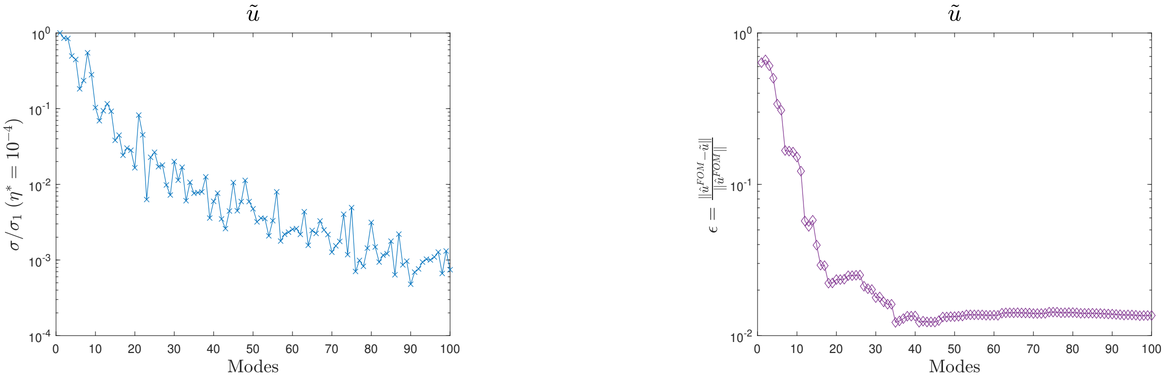
Figure 6.8: 2 Parameter shell generation - simple approach

where the $z$ components of the control points, $\mathbf{P}(z_1)$, are:

$$\begin{bmatrix} P_{00} & P_{01} & P_{02} \\ P_{10} & P_{11} & P_{12} \\ P_{20} & P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} 0 & z_2 & 0 \\ z_2 & z_1 & z_2 \\ 0 & z_2 & 0 \end{bmatrix} \tag{6.3}$$

The parametric domain was set as $\Omega_1 \in [1;8]$ and $\Omega_2 \in [3;6.5]$. For this, $n_1 = 40$ and $n_2 = 20$ uniformly distributed elements were considered throughout their dimensions.

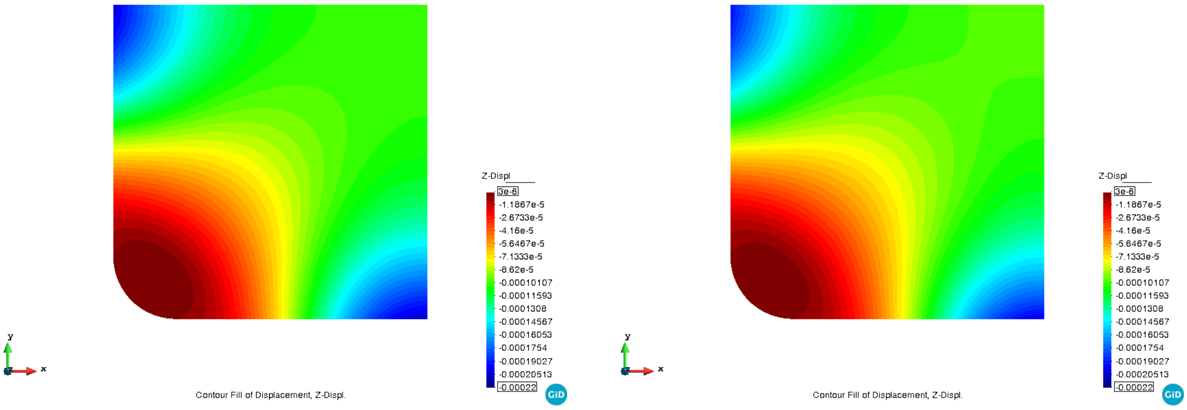
In terms of the separation of the stiffness and force tensors, less than 40 modes were necessary to separate each of them, at a cost of an error of magnitude $10^{-7}$, as it shown on Appendix A.5. The separated tensors were rearranged and the parametric system of linear equations is solved. The PGD proceeds to deliver an approximation of the nodal displacements. These stage´s results in terms of displacements and stresses were compared to the FOM´s, and displayed from Figure 6.9 to 6.13.



(a) Relative importance of each mode - $\tilde{\mathbf{u}}$

(b) Modal update of the relative error mode - $\tilde{\mathbf{u}}$

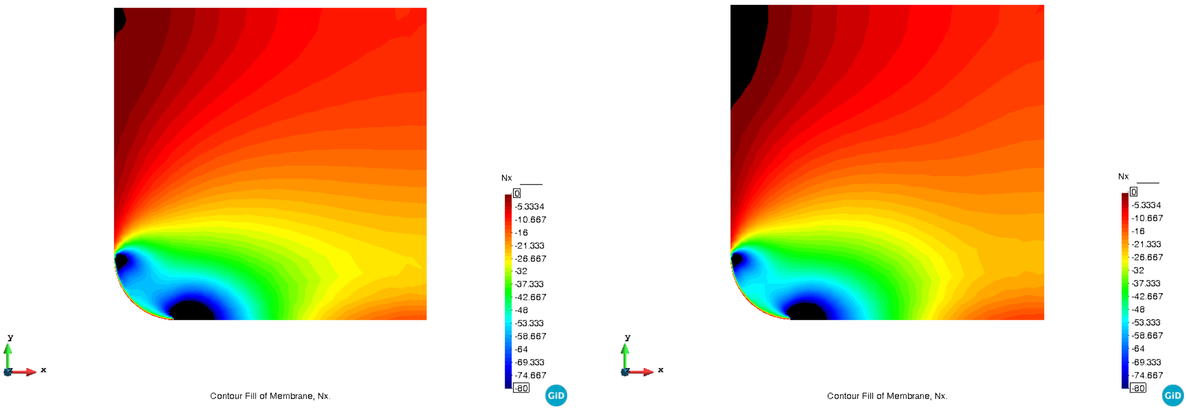
Figure 6.9: Modal enrichment of $\tilde{\mathbf{u}}$ and its error: $z_1 = 8m$, $z_2 = 3m$



(a) $u_z - (m)$

(b) $\tilde{u}_z^{M=35} - (m)$

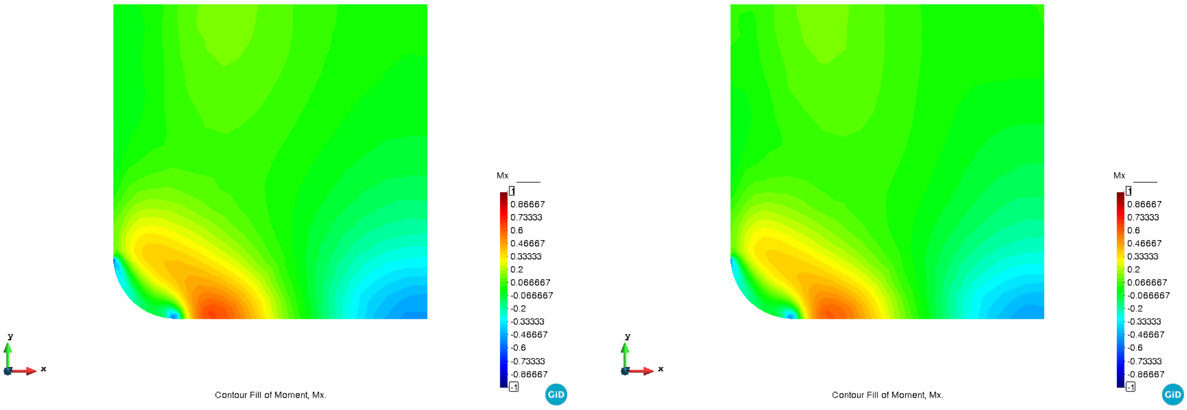
Figure 6.10: z displacements for: $z_1 = 8m$ and $z_2 = 3m$



(a) $N_{xx} - (kN/m)$

(b) $\tilde{N}_{xx}^{M=35} - (kN/m)$

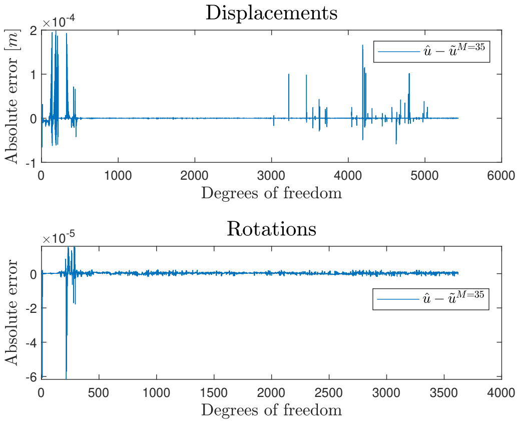
Figure 6.11: $N_{xx}$ stresses for: $z_1 = 3$ and $z_2 = 3m$



(a) $m_{xx} - (kNm/m)$

(b) $\tilde{m}_{xx}^{M=35} - (kNm/m)$

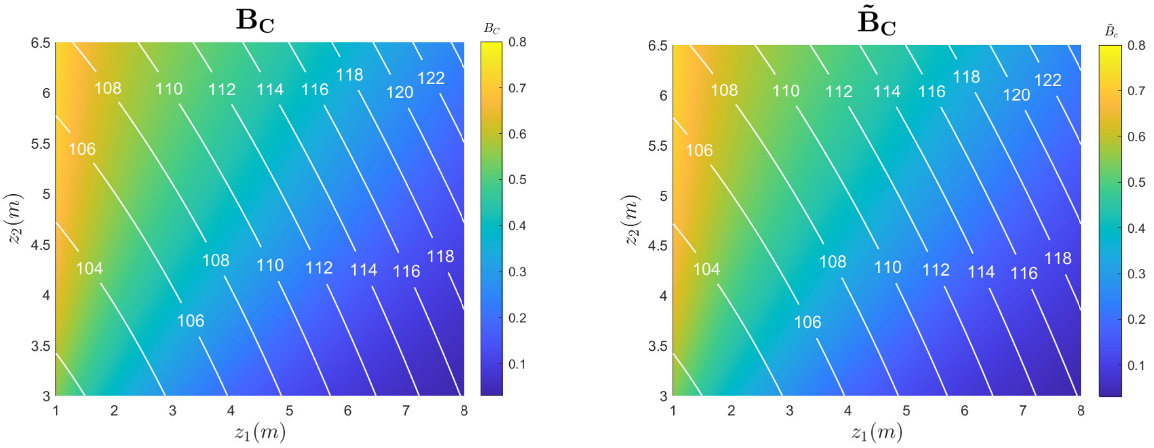
Figure 6.12: $m_{xx}$ for: $z_1 = 8m$ and $z_2 = 3m$



(a)

Figure 6.13: Absolute error of $\tilde{u}$ for: $z_1 = 8m$ and $z_2 = 3m$

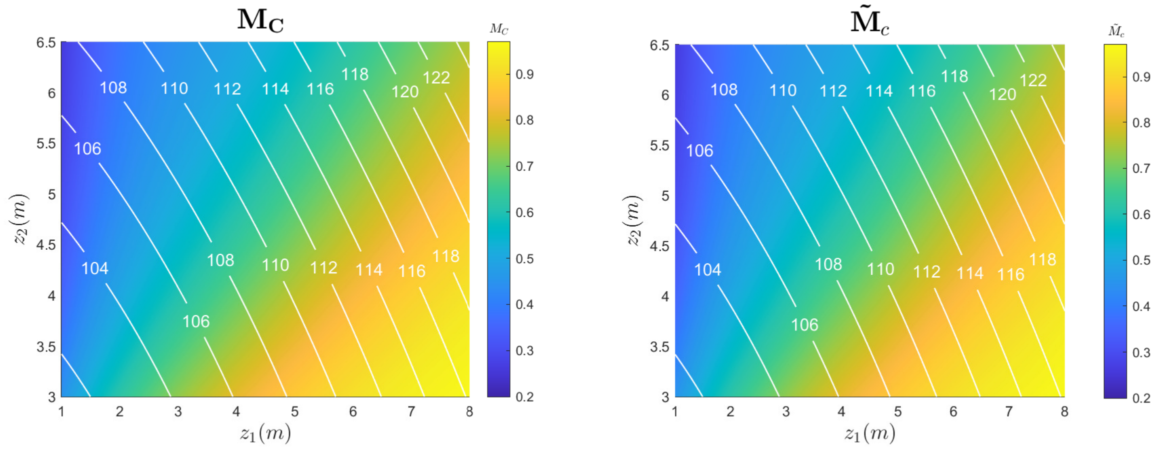
As it can be seen, the PGD approximation seems to be fairly accurate for $\tilde{\mathbf{u}}$. Starting from $M = 35$ the relative error converges and the gain from modal enrichment is negligible, Figure 6.9. Therefore, 35 modes will be adopted for the approximation. Regarding the absolute error, the vast majority of the approximations present a small error, Figure 6.13, that leads to the conclusion that this step was, overall, successful. One may argue that are some outliers among the points shown, however, taking into consideration all the previous graphs, it is assumed that the error will not lead to noticeable loss of quality. The parameters correspond to the combination which is bound to be considered optimal. Taking that into consideration, for two parameters and for a simple Bézier Surface formulation, the model is unable to surpass the efficiency, in terms of bending behaviour, of the optimal combination in the previous case. In terms of bending moments, both solution present similar maximum values,despite having different geometric configurations. Only 5 modes were necessary to describe the coefficients in a separated format. In terms of stresses and bending moments, the FOM and ROM provide similar results, Figures 6.11 and 6.12, with the ROM assuming slightly more traction in the borders. The FOM´s and ROM´s energy coefficients are illustrated on Figures 6.14 and 6.15:



(a)

(b)

Figure 6.14: FOM and ROM bending coefficients with shells´ area isolines - 2 parameters(simple)



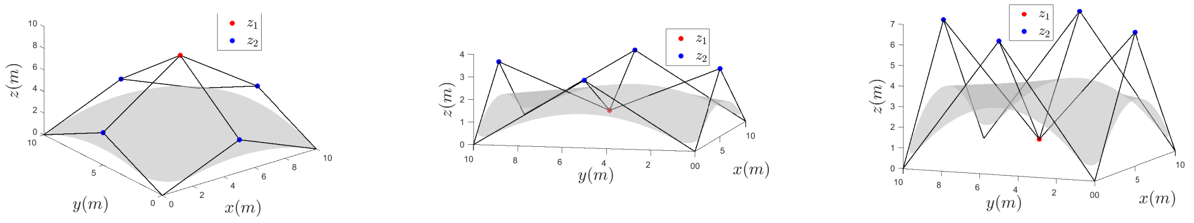
(a)

(b)

Figure 6.15: FOM and ROM membrane coefficients with shells´ area isolines - 2 parameters(simple)

The PGD was not only successful to create a ROM for the coefficients, but presented satisfactory results when creating a ROM for stresses and displacements. For the latter two, the savings in terms of memory usage are extremely relevant. To store similar results, the ROM requires around $\frac{35\times(ndof+n_1+n_2)}{ndof\times n_1\times n_2}$ = $4\%$ of the FOM´s memory.

However, the results in terms of form-finding are dull and aesthetically monotonous, see Figure 6.3. The optimal combination is located at the corner of the parametric domain, see Figures 6.14 and 6.15, if the domain would be extended, the optimal combination would remain on the corner of the new parametric domain. Thus, cancelling the advantage of creating a FOM and ROM. Nevertheless, the domain cannot be extended, as the geometry would start to lose physical significance. Therefore, new ways of generating the geometry of the shell were considered. Firstly, polynomials of degree 4 will be implemented to conceive new shapes, secondly, trigonometrical functions will be considered to define the z-component of the control points. These new considerations were already applied on Section 6.1 and provided satisfactory outcomes. Now, the application will be tested for 2 parameters.



(a) Optimal solution: $z_1 = 8m$ and $z_2 = 3m$

(b) $z_1 = 1m$ and $z_2 = 3m$

(c) $z_1 = 1m$ and $z_2 = 6.5m$

Figure 6.16: Geometries corresponding to the admissible corners of the domain - 2 parameters(simple)

## 6.3 Shell defined by 2 parameters - extensive approach

### 6.3.1 Generation of the geometry

To obtain new shapes, lower bending coefficients and less traction a new approach will be followed, relatively to the previous section by adapting the Bézier formulation on Section 6.1, as in Figure 6.17.

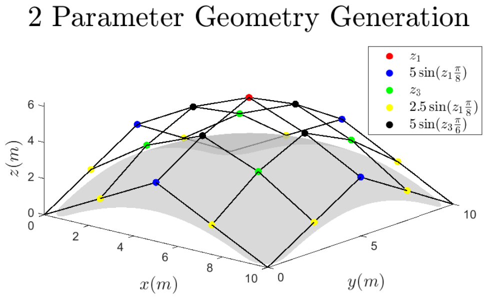


Figure 6.17: 2 parameter shell generation - extensive

The analytical expression of the surface corresponds to the one lecture in Section 5.1. The $P(z_1, z_3)$ matrix is formally defined as:

$$\begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} & P_{04} \\ P_{10} & P_{11} & P_{12} & P_{13} & P_{14} \\ P_{20} & P_{21} & P_{22} & P_{23} & P_{24} \\ P_{30} & P_{31} & P_{32} & P_{33} & P_{34} \\ P_{40} & P_{41} & P_{42} & P_{43} & P_{44} \end{bmatrix} = \begin{bmatrix} 0 & D & B & D & 0 \\ D & C & E & C & D \\ B & E & A & E & B \\ D & C & E & C & D \\ 0 & D & B & D & 0 \end{bmatrix} \tag{6.4}$$

with $A = z_1$, $B = 5\sin(z_1\frac{\pi}{8})$, $C = z_3$, $D = B/2$, $E = 5\sin(z_3\frac{\pi}{6})$. The control points expressions were chosen to circumvent the problems identified in the previous Section.

### 6.3.2 Parametric Domain

The parametric domain will be established solely by the principles mentioned in Section 5.2, by setting limits to the maximum height of the shell. The height of the shell must be controlled, not only due to physical restraints, but also because of the model´s fragilities. The solution provided must be realistic, and the model must be be accurate, if the height is bound to significantly increase, the ”inflation” simplification would become compromised and the $6^{th}$ degree of freedom, the drilling, would no longer be negligible. However, respecting this guideline, while generating a constant size tensor, is impossible, for $D > 1$. For instance, for $D = 2$, $int(\Omega)$ would not resemble a square, but a triangle. However, these guidelines are not mandatory and non-desired parameter combinations can be excluded in post-processing.

A viable way to overturn this problem, would be to only use combinations that fit the mentioned criteria, by compacting the desired combinations from the stiffness tensor of order 3, for example, into a matrix. Another way to solve the problem would be to perform a mapping operation. For instance, for $D = 2$, the admissible region of parameters is a triangle and the compatible region is rectangular, therefore a mapping between these two regions can be performed.

However, for $D > 3$, the latter solution lacks visual representation and may not be possible. Therefore, the $1^{st}$ and $2^{nd}$ option are assessed. Although, the second solution may seem more suitable, as it requires less points, the resulting parametric domain may not be convex. For this reason and for using more points, the first approach may lead to a better PGD approximation.

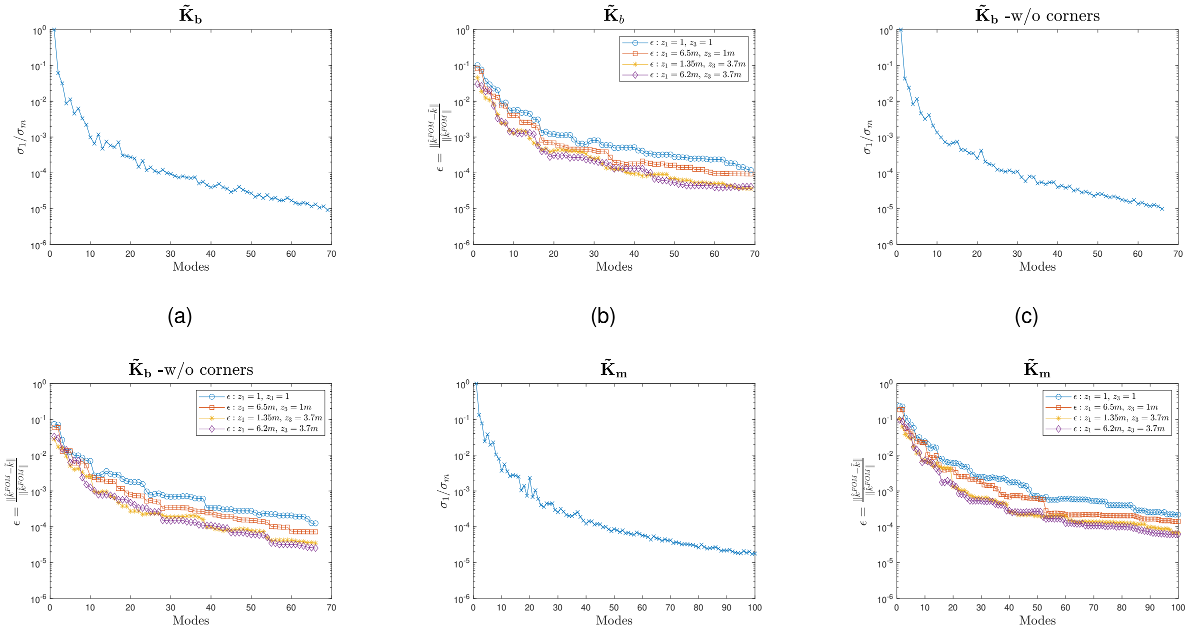
In this sense $\Omega_1 \in [1;8]$ and $\Omega_2 \in [1;4]$ with $n_1 = 40$ and $n_2 = 20$ uniformly distributed elements throughout their dimensions. The coefficients used to define the z component of the control points are directly linked with the parametric domains´s geometrical boundaries, and were slightly increased to face the increased error, normally, observed on the parametric domains´s boundaries.

Relatively to the previous case, the parametric domains were set to be similar in size and content. The main goal of this decision is to assess the gains and limitations of using a more creative approach to form-finding.

### 6.3.3 Separation of the Stiffness tensors

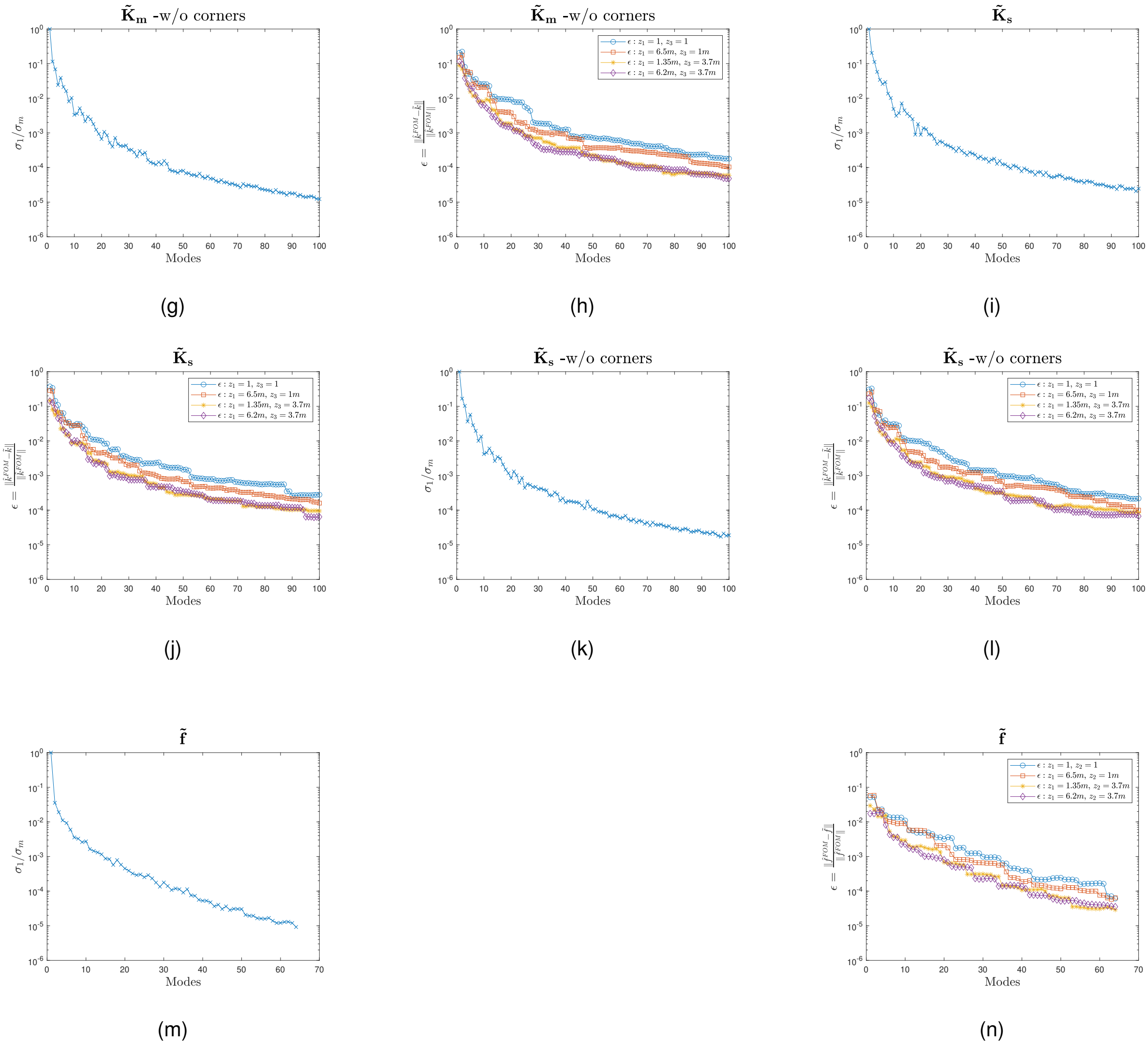
The strategy is to perform the PGD Tensor Separation on the Stiffness tensors in the multidimensional format, as illustrated in Figure 5.9 and explained on Section 5.4. The separation is carried on tensors, that, for each combination carry a rank one tensor, with the non-zero elements of their original stiffness matrix. The error associated with this step will be computed according to equation (5.28), considering modal enrichment.

As a prompt approach to evaluate it, the error associated with the separation of the tensors will be assessed for four parameter combinations, whose geometries are illustrated on Figure 6.28. It is important to recall that, in addition, the Stiffness tensors that do not consider the effect of the corners (singularities) will also be separated, so the energies can be more accurately calculated on the following steps. The following results were obtained:



(a) (b) (c)

(d) (e) (f)



(g) (h) (i)

(j) (k) (l)

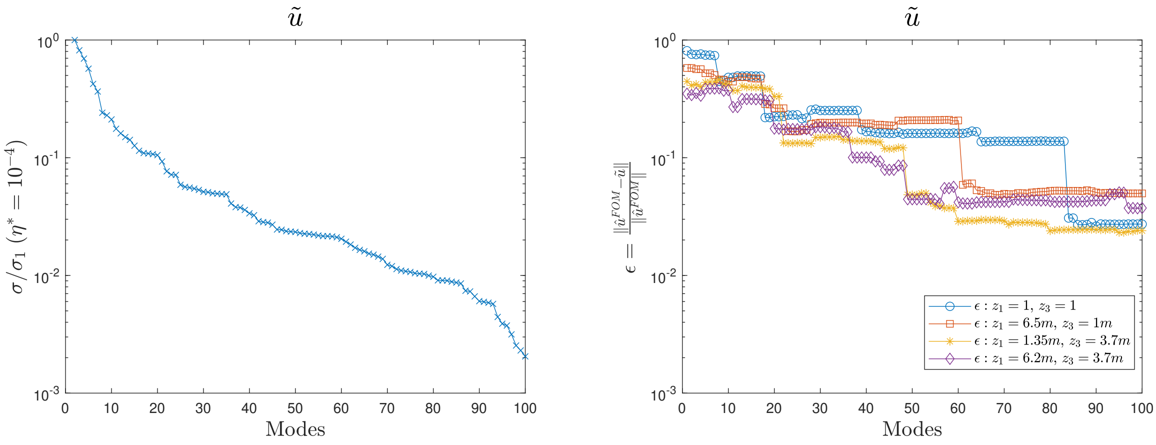
(m) (n)

Figure 6.18: Stiffness tensors and force tensor separation - 2 parameters (extensive)

Contrary to the previous case, the separation met its end by reaching the maximum number of prescribed modes. By introducing trigonometric functions on the definition of the control points and by having two parameters, the model is expected not to be strictly monotonous: to have ”hills” (maximums) and ”valleys” (minimums) inside its domain. For those reasons, it may be more challenging for the PGD to separate the input data. Two of the parameter combinations introduced are located at the core of its domain. The reason for these addition was to compare the error between combinations located at the boundaries and inside of the parametric domains, that is, $z_i \in int(\Omega_i)$. It is perceptible that, when on the interior of the parametric domain, the separation, for a given accuracy, is faster than for combinations at the borders of the parametric domains, Figure 6.18. The present separation relative error is roughly 100 times superior than for the previous case.

### 6.3.4 Solving the parametric linear system of equations

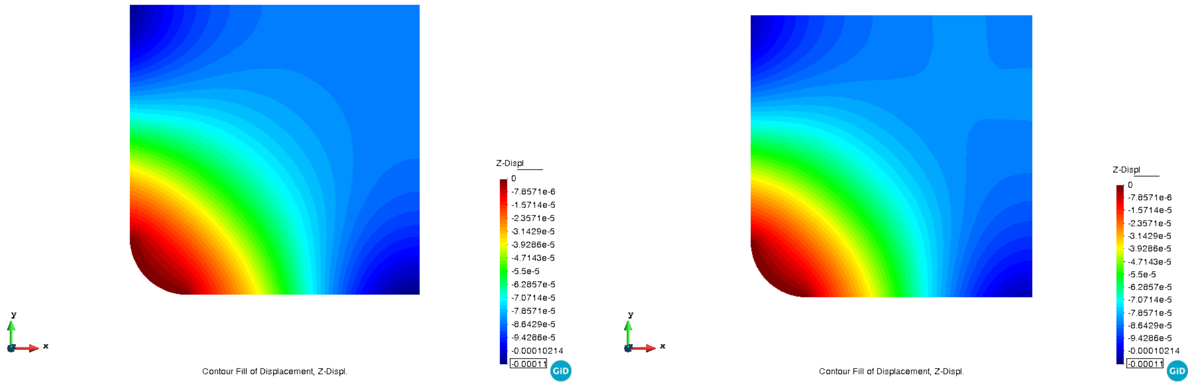
The separated tensors were divided into their unparametric and parametric components, as described in the previous chapters. The relative error was evaluated for the same 4 combinations.



(a)

(b)

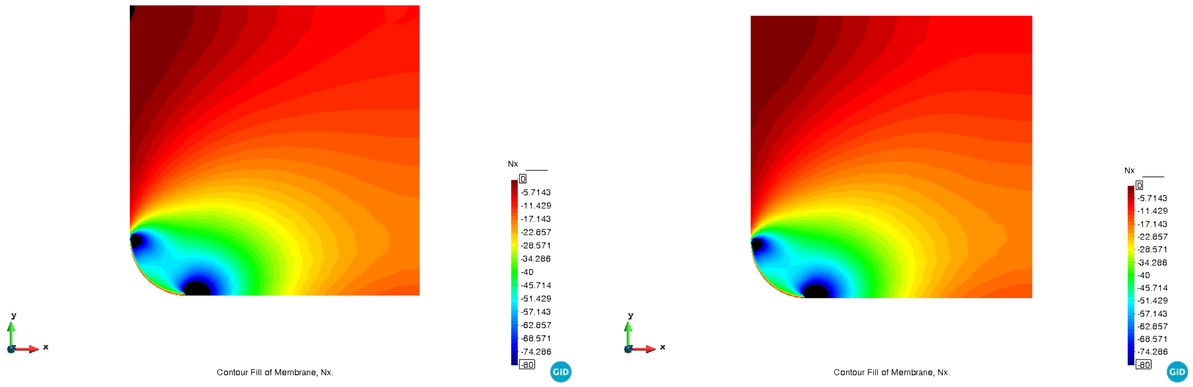
Figure 6.19: Modal enrichment of $\tilde{\mathbf{u}}$ and its error - 2 parameters (extensive)



(a) $u_z - (m)$

(b) $\tilde{u}_z - (m)$

Figure 6.20: z displacements for: $z_1 = 6.2m$ and $z_3 = 3.7m$

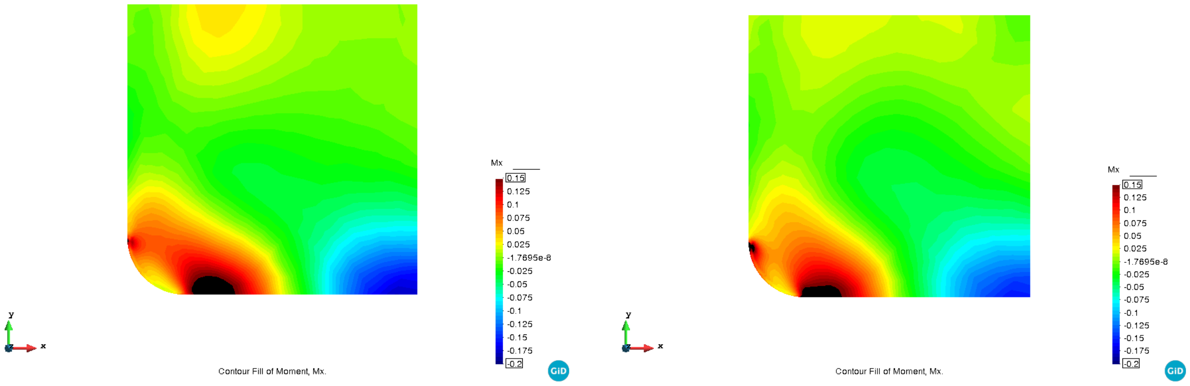


(a) $N_{xx} - (kN/m)$

(b) $\tilde{N}_{xx} - (kN/m)$

Figure 6.21: $N_{xx}$ stresses for: $z_1 = 6.2m$ and $z_3 = 3.7m$

For the optimal solution, neither significant traction nor moment are found on the shell. This translates in an improvement relatively to the previous trials. As for the PGD, despite the lower quality of the separated stiffness tensors, the error associated with $\tilde{\mathbf{u}}$ is surprisingly similar to the one in Figure 6.9.



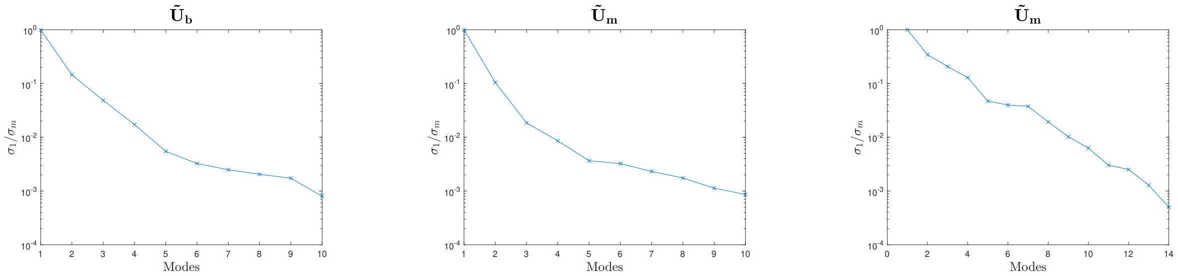
(a) $m_{xx} - (kNm/m)$ (b) $\tilde{m}_{xx} - (kNm/m)$

Figure 6.22: $m_{xx}$ for: $z_1 = 6.2m$ and $z_3 = 3.7m$

Thus, the approximations of $\tilde{\mathbf{u}}$, $\tilde{\mathbf{N}}_{\mathbf{xx}}$ and $\tilde{\mathbf{m}}_{\mathbf{xx}}$ are also accurate and reasonable, as seen in Figures 6.20, 6.21 and 6.22. The optimal solution presents 5 times less bending, in absolute value, than the one provided by 1 parameter, in Figure 6.5.

### 6.3.5 Computing the approximation of the energies and energy coefficients

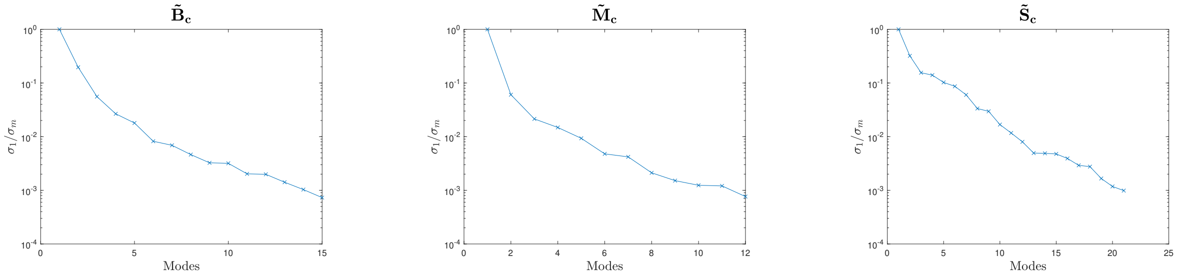
As mentioned, to reach a higher quality of results, only the stiffness tensors disregarding the corners´contribution were used to calculate the energies. The energies were calculated following equations (5.20), (5.21), (5.22) and (5.23), and then truncated and compressed, Figure 6.23.



(a) 962516 modes truncated (96.3 %) (b) 913572 modes truncated (91.4 %) (c) 875294 modes truncated (87.5 %)

Figure 6.23: Modal enrichment of separated energy component tensors

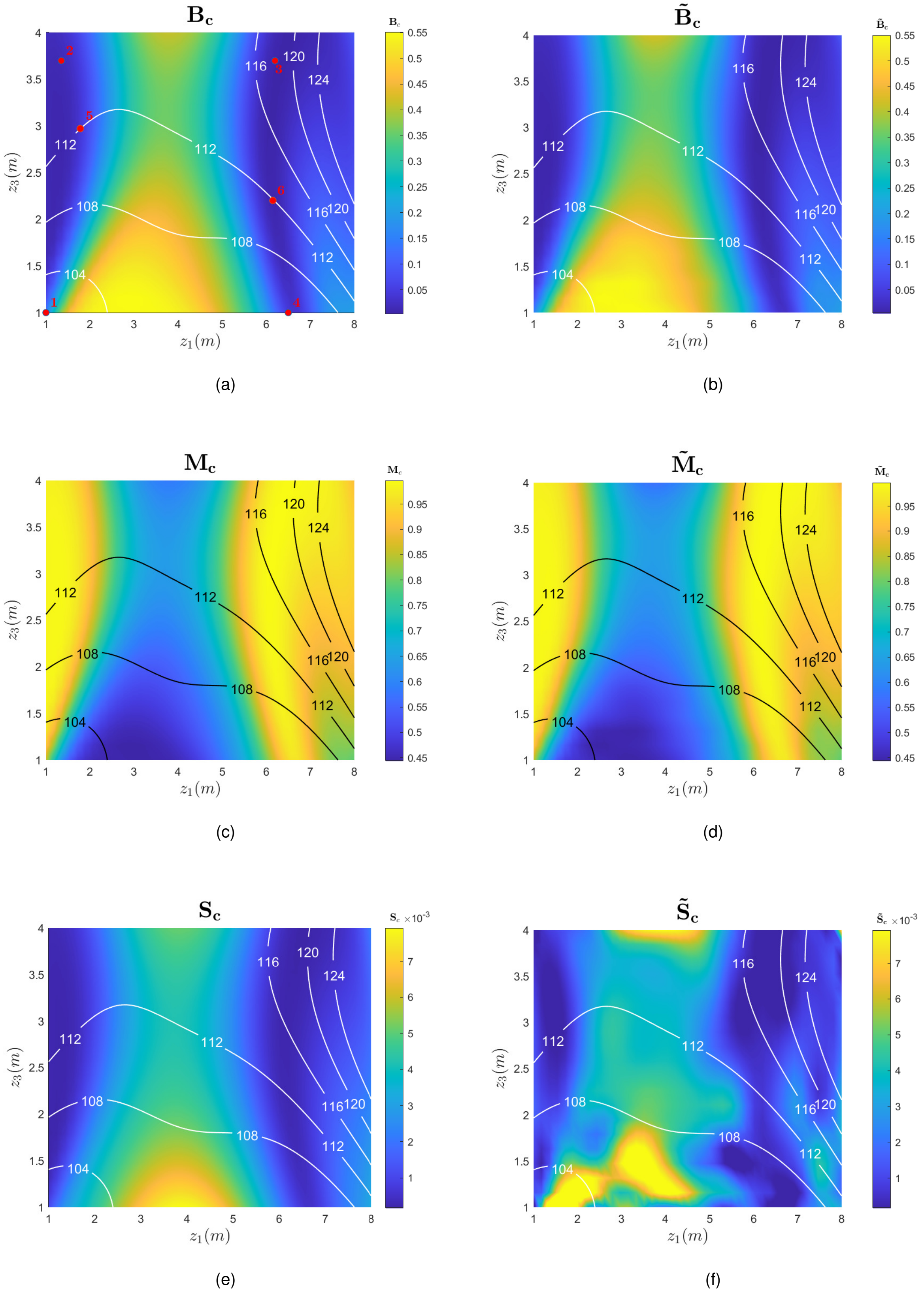
The Hadamard Division is performed to compute the energy coefficients, Figure 6.24.



(a) (b) (c)

Figure 6.24: Modal enrichment of separated energy coefficient tensors

Finally, the results provided by the Full Order Model and the Reduced Order Model, built with PGD, regarding the energy coefficients are displayed, as well as the shell´s area isolines, Figure 6.25 .



(a)

(b)

(c)

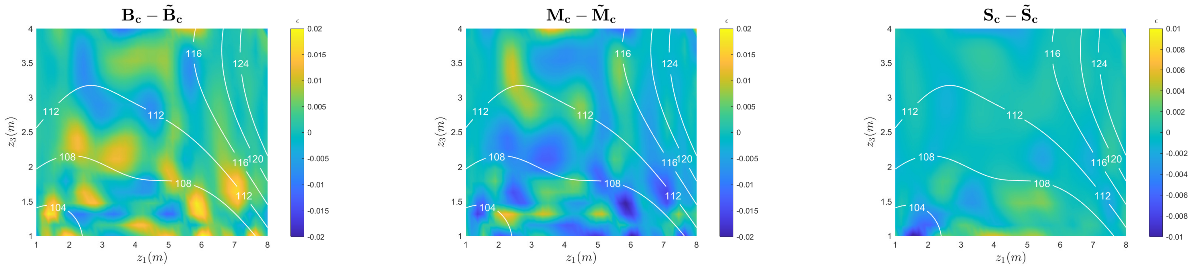
(d)

(e)

(f)

Figure 6.25: Energy coefficients provided by the FOM (left) and ROM (right) - 2 parameters extensive

The absolute error, for each approximation, are illustrated with the isolines for the areas as:



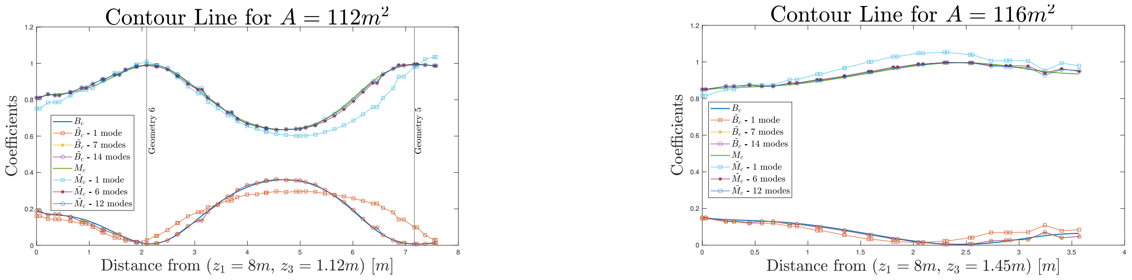
(a)

(b)

(c)

Figure 6.26: ROM´s energy coefficients absolute error

The surface representing $\mathbf{B_C}$ is no longer monotonous, despite this, the ROM was able to reproduce and accurately choose the most suitable configuration that minimizes bending behaviour, Figure 6.25. The error map in Figure 6.26 indicates that the absolute error is proportional to the FOM´s $B_C$, presenting an acceptable maximum error of $0.01$, and almost 0 for $B_C < 0.01$.

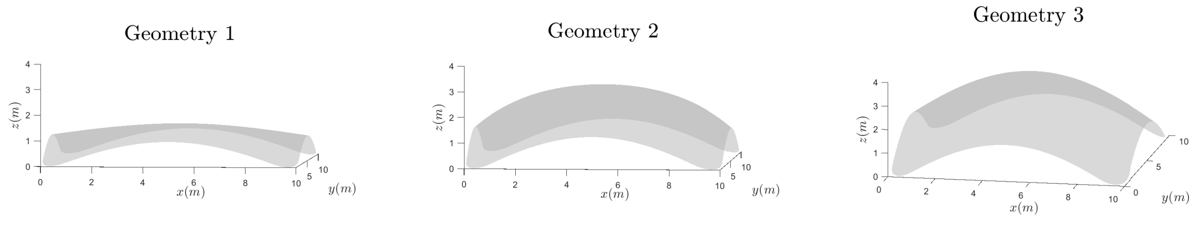


(a)

(b)

Figure 6.27: ROM and FOM energy coefficients for $A = 112m^2$ and $A = 116m^2$

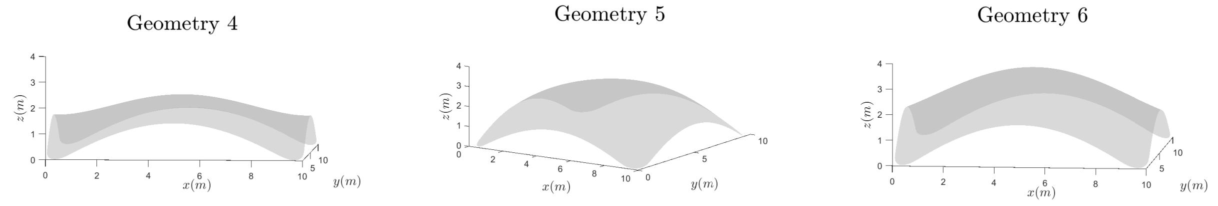
From the area isolines depicted in Figure 6.27, it is perceptible that half of the modes would be sufficient to deliver an accurate approximation for both energy coefficients. Various configurations in this formulation can lead to structures with bending coefficients lower than $0.01$, and practically be considered optimal. For these geometries, focusing on $A$ can help choose the fittest configuration. For instance, for $A = 112m^2$ and $A = 116m^2$, 3 possible configurations can be chosen, Figure 6.27, hence, factors such as aesthetics and costs become more relevant. Even though material savings for $L = 10m$ are irrelevant, if the span would be more realistic, $L = 50$, optimizing $A$ would also be a top priority.



(a) $z_1 = 1, z_3 = 1, B_C = 0.14$

(b) $z_1 = 1.35, z_3 = 3.7, B_C = 0.016$

(c) $z_1 = 6.2, z_3 = 3.7, B_C = 0.0035$



(d) $z_1 = 6.5, z_3 = 1, B_C = 0.037$   (e) $z_1 = 1.5, z_3 = 2.8, B_C = 0.0.0063$   (f) $z_1 = 6.2, z_3 = 2.2, B_C = 0.00697$

Figure 6.28: Geometries generated with trigonometrical functions and $n = 4$

This creative approach allowed to reach its three objectives, mentioned in 6.1. Traction free shell were generated, Figure 6.21, the $B_C$ was reduced $\approx$ 10 times with the introduction of a new parameter, and new shapes were found. In conclusion, this formulation delivers dozens of optimal solutions ($B_C < 0.01$), that consume less material than in previous cases, while generating more aesthetically diverse types of surfaces, as displayed in Figure 6.28. The geometries are marked in Figure 6.25 **(a)**.

## 6.4 Shell defined by 3 parameters

At this stage, the reader must be familiar with the followed methodology, therefore, only relevant topics will be presented , for this case.

### 6.4.1 Generation of the geometry

The shell´s geometry is described by three parameters, $z_1$,$z_2$,$z_3$ as it was already seen in Section 5.3.1, and it is analytically defined as in Figure 6.29.

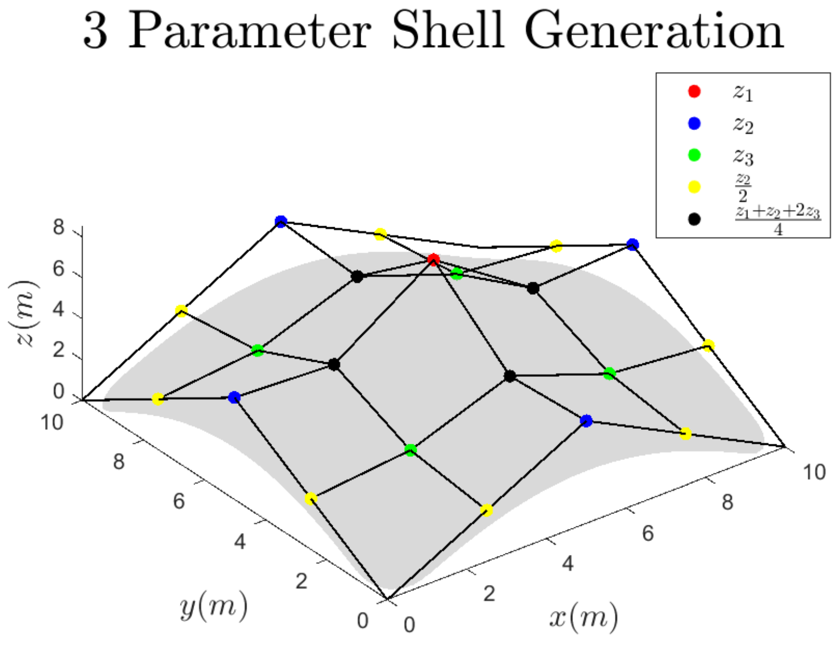


Figure 6.29: 3 parameter shell generation

where the $z$ components Bézier control points, $\mathbf{P}$, are:

$$\begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} & P_{04} \\ P_{10} & P_{11} & P_{12} & P_{13} & P_{14} \\ P_{20} & P_{21} & P_{22} & P_{23} & P_{24} \\ P_{30} & P_{31} & P_{32} & P_{33} & P_{34} \\ P_{40} & P_{41} & P_{42} & P_{43} & P_{44} \end{bmatrix} = \begin{bmatrix} 0 & D & B & D & 0 \\ D & C & E & C & D \\ B & E & A & E & B \\ D & C & E & C & D \\ 0 & D & B & D & 0 \end{bmatrix} \tag{6.5}$$

with $A = z_1$, $B = 2_2$, $C = z_3$, $D = A/2$, $E = (A + 2C + B)/4$.

### 6.4.2 Parametric Domain

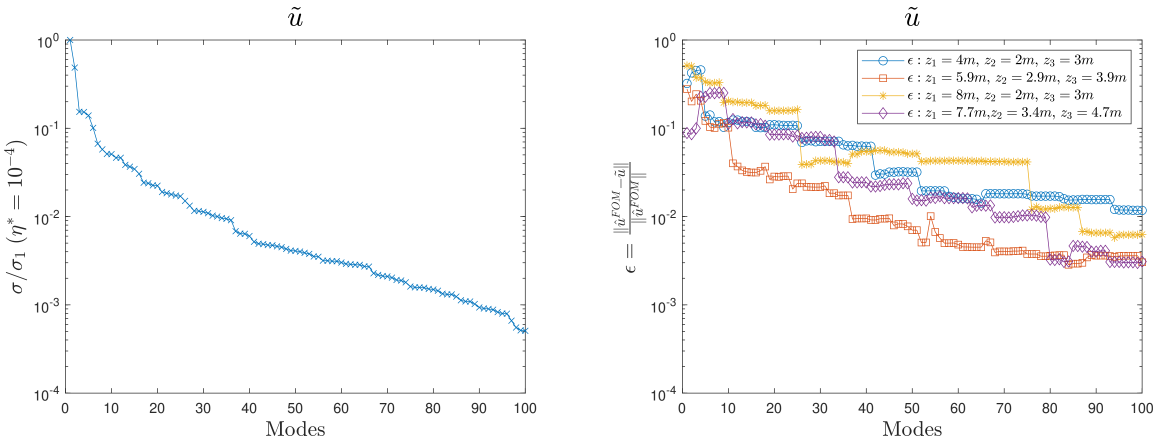
By now, the problem has been exhaustively studied for a large number of parameters and combinations, thus, a general idea of the parameters needed to reach an almost pure membrane behaviour has been acquired. For this reason, the parametric domains was established with a rough discretization, with $\Omega_1 = [4, 8], \Omega_2 = [2, 4], \Omega_3 = [3, 5], n_1 = 16, n_2 = 8, n_3 = 8$, forming 1024 combinations, more or less, with a little more combinations than the previous case.

### 6.4.3 Separation of the Stiffness matrices

For this case, the methodology is the same as used. Nevertheless, as an attempt to use fewer modes on the separated stiffness tensors, $\eta^*$ was set to $10^{-5}$. For this formulation, it is noticeable that, even with 3 parameters, the separation illustrated on Appendix A.3, for a given accuracy, has a faster convergence than for the case in Section 6.3, depicted on Figure 6.18. This may lead to the following conclusions: Adding parameters may be more efficient than using trigonometric functions to define the z component of the control points. A threshold $\eta^* = 10^-6$, may have been too conservative, nevertheless, it produced fair results.

### 6.4.4 Solving the parametric linear system of equations

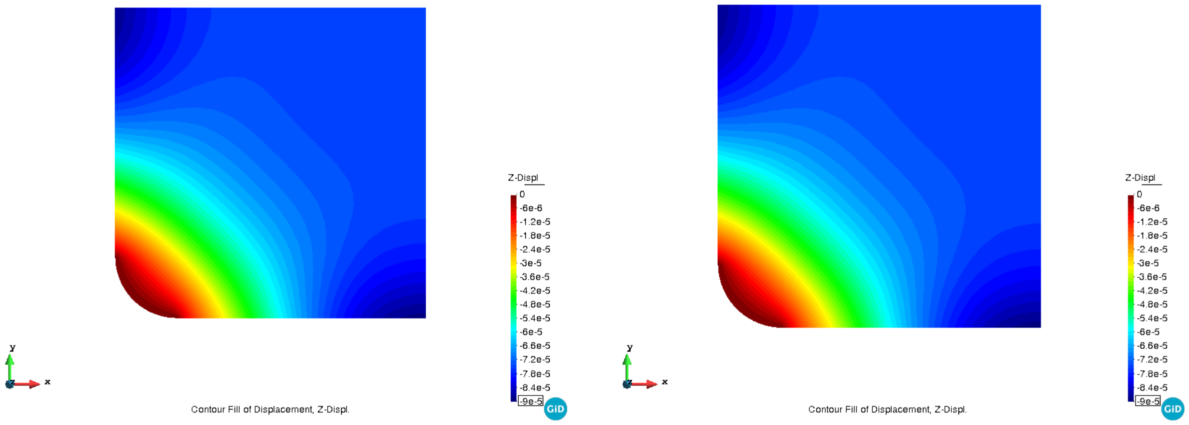
The observation made in the last paragraph is now confirmed. The present formulation eases the separation process, as for the combinations considered, $u$ presents a relative error 10 times inferior to the previous case. This linear formulation leads to an accurate approximation of the FOM. As more parameters are added, the shape has more freedom to assume geometries that reduce the bending behaviour, decrease maximum bending moment and z displacements. The magnitude of the normal stresses remains unchanged. The Bézier Surface formulation is extended, the closer the geometries are to an antifunicular configuration. Despite already having found configurations that, practically, behave as pure membranes.



(a) $u_z(m)$

(b) $\tilde{u}_z(m)$

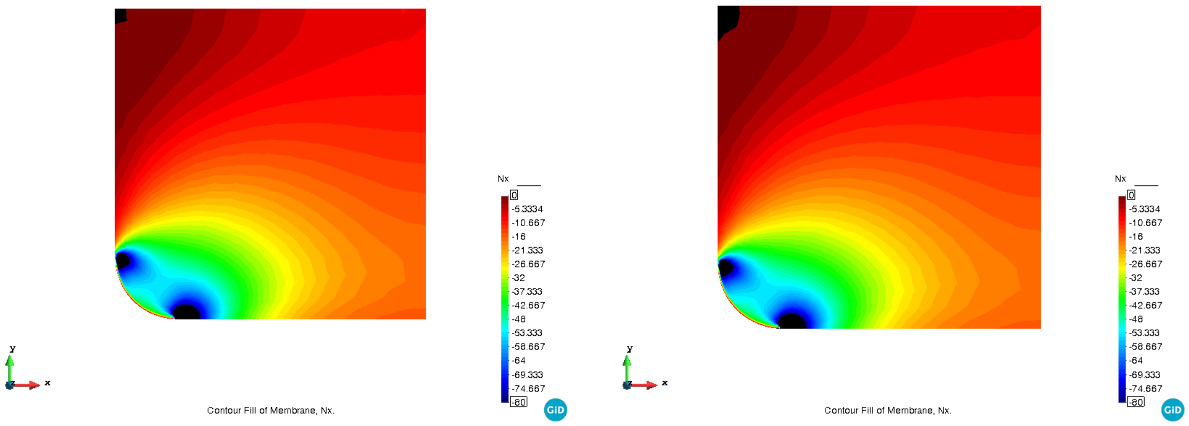
Figure 6.30: Modal enrichment of $\tilde{\mathbf{u}}$ and its error



(a) $u_z(m)$

(b) $\tilde{u}_z(m)$

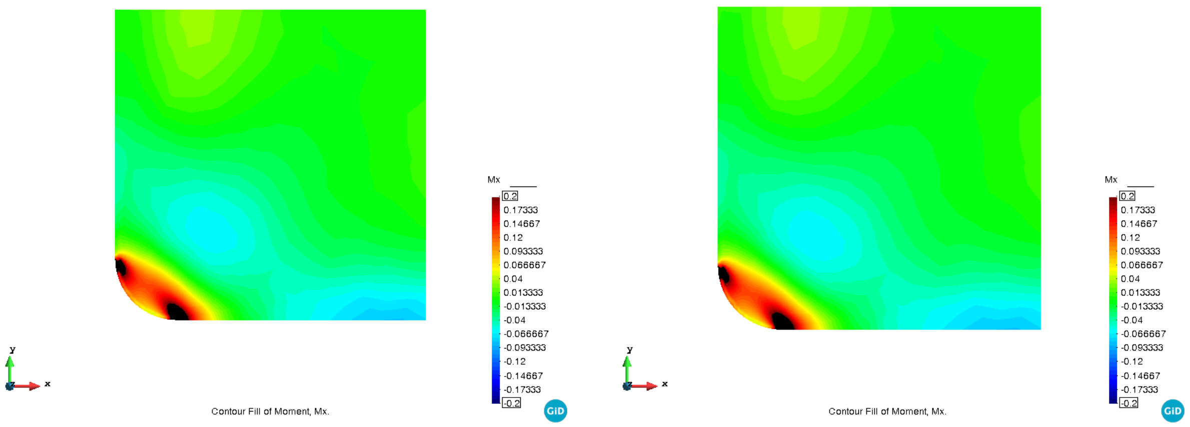
Figure 6.31: z displacements for: $z_1 = 7.7m$, $z_2 = 3.4m$ $z_3 = 4.7m$



(a) $N_{xx} - (kN/m)$

(b) $\tilde{N}_{xx} - (kN/m)$

Figure 6.32: $N_{xx}$ stresses for: $z_1 = 7.7m$, $z_2 = 3.4m$ $z_3 = 4.7m$



(a) $m_{xx} - (kNm/m)$

(b) $\tilde{m}_{xx} - (kNm/m)$

Figure 6.33: $m_{xx}$ for: $z_1 = 7.7m$, $z_2 = 3.4m$ $z_3 = 4.7m$

### 6.4.5 Computing the approximation of the energies and energy coefficients

Similarly to previous cases, only a few modes were needed for the PGD approximation to deliver quite accurate results.

As the coefficients are now a function of $z_1$, $z_2$ and $z_3$, it is challenging to visualize the results without having the burden of losing information. For this reason, the results are firstly shown directly for every combination to validate the ROM.

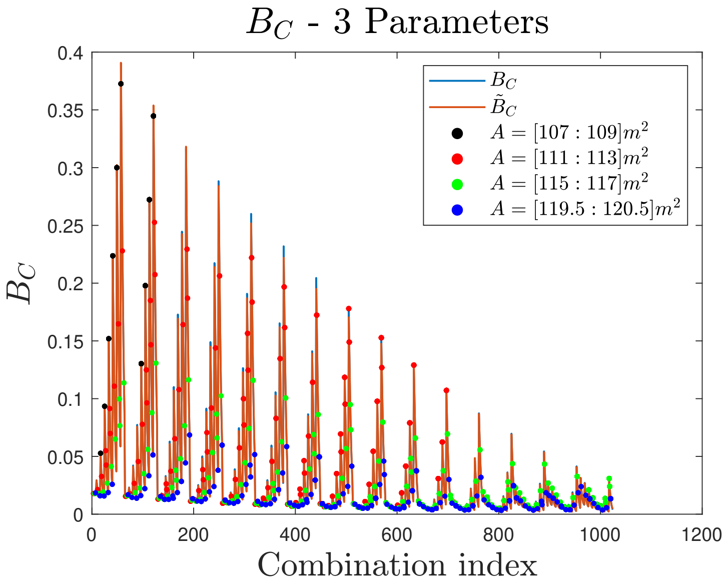


Figure 6.34: ROMs and FOMs bending coefficient and approximated area for the 3 parameter configuration

As in the previous case, the PGD provides reasonable results for the chosen parametric domain. The maximum absolute error is around $0.015$, which is acceptable, as its occurs when $B_C$ peaks and is above $0.10$. The relative error also indicates, that for the most part, the approximation carries an error $5\%$, which is also reasonable. Additionally, as the $B_C$ starts to significantly decrease, this metric

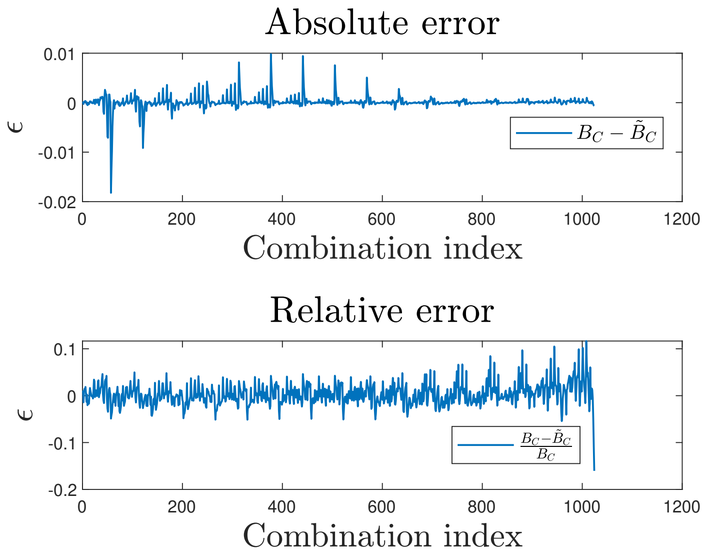
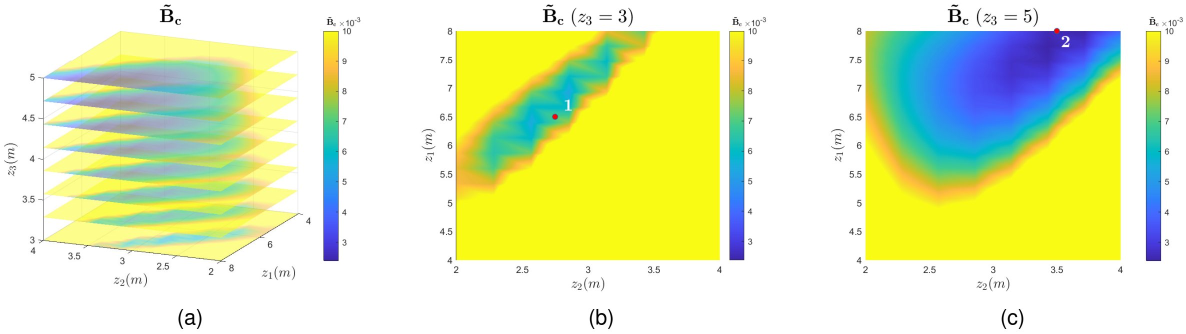


Figure 6.35: Relative and absolute error for the 3 parameter configuration

becomes biased, as the relative error begins to rise. The membrane coefficient was chosen not to be presented as $M_C = 1 - B_C$, for a negligible shear behaviour. For a given combination, the parameters can be easily found, however, it requires an additional step to interpret the data. The areas were also included in the primary analysis of the results, for an Area, $A = 108m^2$ (dark markers) the membrane behaviour is not promising. As for $A = 112m^2$ (red markers), there are some combinations that present the sought behaviour, but for this formulation, this area is not particularly expressive in terms of number of combinations. For $A = 116m^2$ (green markers) and $A = 120m^2$ (blue markers), most of the combinations generate a shell whose membrane behaviour is above $95\%$. Note: Due to the rough discretization of the parametric domain, finding the combinations that would result in a scalar area, would be impossible, so, small ranges for the area were set.

Although the first approach to visualize the results gave valuable insights to what to explore next, to obtain a better perception of the results, the parametric domain has to be sliced. The ROM´s bending coefficients were studied for the third sectional domain boundaries, $z_3 = 3m$ and $z_3 = 5m$, and for $A = 116m^2$ and $A = 120m^2$, as they deliver more interesting results, in terms of approaching the a pure membrane behaviour. Not only that, but also it will be easier to directly compare to the results obtained on the previous cases.



(a)

(b)

(c)

Figure 6.36: Slices of $\tilde{\mathbf{B}}_\mathbf{C}$ distributed throughout $\Omega_3$

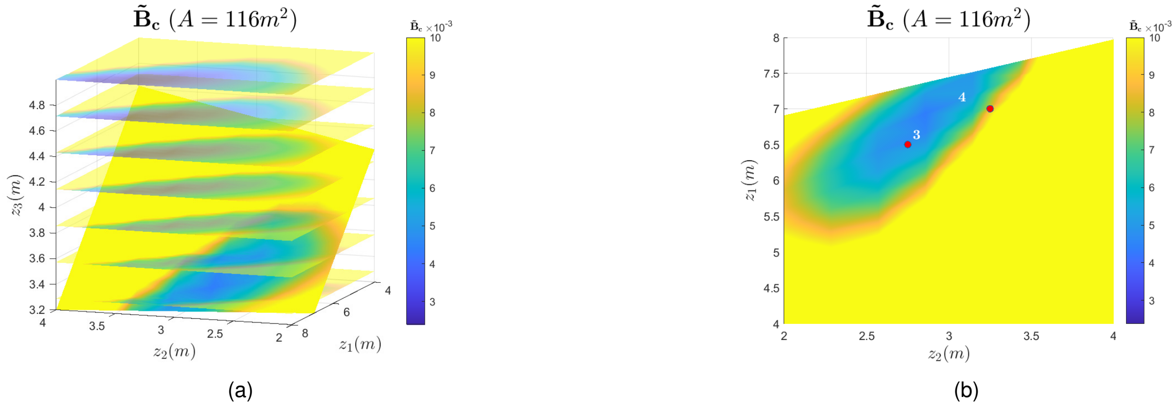


Figure 6.37: $B_C$ slices for $A = 116m^2$

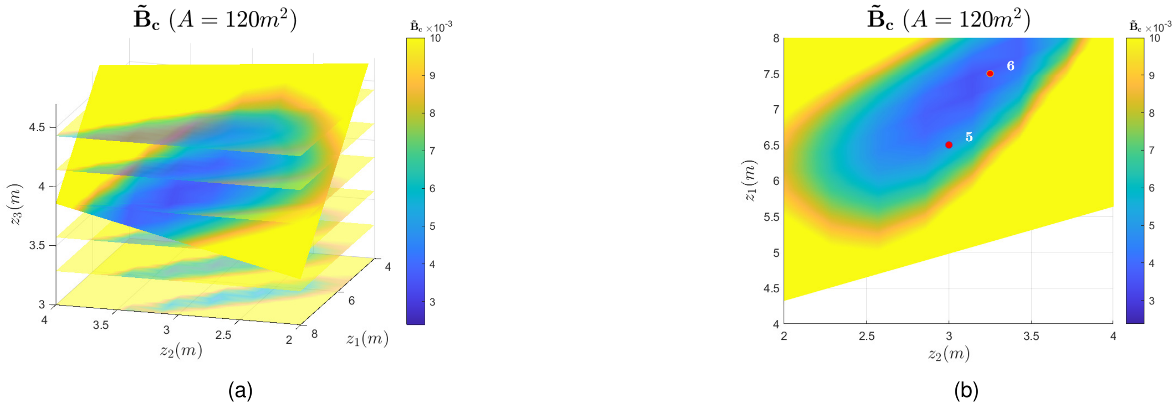
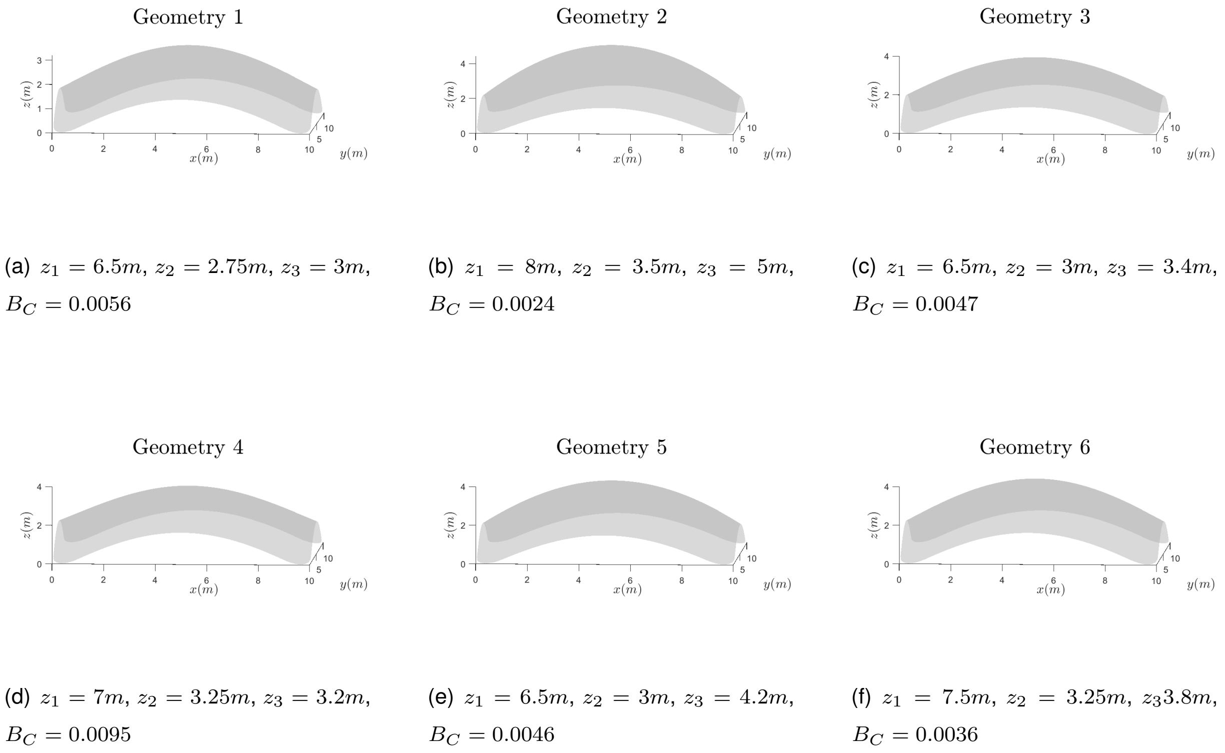


Figure 6.38: $B_C$ slices for $A = 120m^2$



(a) $z_1 = 6.5m$, $z_2 = 2.75m$, $z_3 = 3m$, $B_C = 0.0056$

(b) $z_1 = 8m$, $z_2 = 3.5m$, $z_3 = 5m$, $B_C = 0.0024$

(c) $z_1 = 6.5m$, $z_2 = 3m$, $z_3 = 3.4m$, $B_C = 0.0047$

(d) $z_1 = 7m$, $z_2 = 3.25m$, $z_3 = 3.2m$, $B_C = 0.0095$

(e) $z_1 = 6.5m$, $z_2 = 3m$, $z_3 = 4.2m$, $B_C = 0.0046$

(f) $z_1 = 7.5m$, $z_2 = 3.25m$, $z_3 3.8m$, $B_C = 0.0036$

Figure 6.39: Optimal geometries generated - 3 parameters

Overall, $B_C$ is extremely low in the whole domain, thus, the values were set to a maximum of $0.01$ to easily perceive the combinations that lead to optimal shapes. From the displayed sections, it is perceptible that dozens of solutions are available, as seen in Figures 6.36, 6.37 and 6.38, by setting the maximum $B_C$ to $0.01$. Although the generated geometries from the present formulation practically act as pure membranes, they present uninteresting forms, from a form-finding standpoint, mainly due to the formulation and the narrow parametric domain. Not only that, but starting from 3 parameters it is hard to visualise the effect of the variation of one parameter in the shell´s geometry and structural behaviour. Nonetheless, each optimal configuration be slightly optimized, as it will be seen on the next Section.

## 6.5 Heuristic approach

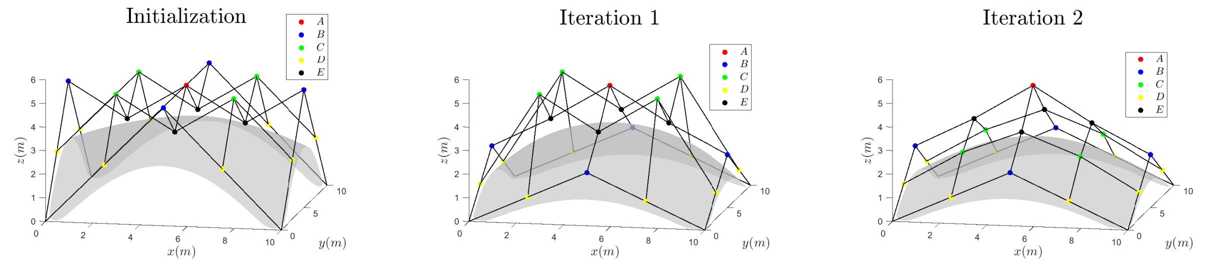
Another way to find an optimal configuration, for any height, is through successive iterations, almost as in a greedy approach. Starting with an arbitrary formulation, with $n = 4$, univariate functions $A$, $B$, $C$, $D = B/2$ and $E$ in terms of $z_1$. An arbitrary $z_1$ is chosen and $A$, $C$ and $E$ are locked to this value. Secondly, $B$ and $D$ are defined in terms of $z_2$, and the optimal $z_2$ is found. Then, $C$ is written in terms of $z_3$ and the optimal parameter is calculated. At last, E is parameterized and the optimal parameters are defined. This process stop until $D$ parameters are fixed or until the bending behaviour of the former structure is less expressive than the latter. This reasoning can be applied to either improve an already established configuration or to find an an optimal configuration from scratch.

For instance, for the optimal configuration in Section 6.4, with $B_C = 0.0056$ $z_1 = 6.5m$, $z_2 = 2.75m$ and $z_3 = 3m$, by fixing the control points $A = 6.5m$, $B = 2.75m$, $C = 3m$ and setting $E = z_5$ with $\Omega_5 = [3; 5]m$ and $n_5 = 50$, the optimal is slightly improved. The new optimal configuration presents $B_C = 0.0050$, which obviously proves how close the previous configuration was from ”true antifunicular shape” . Geometrically, this leads to a small adjustment (due to rough discretization) of $E$ to be $4m$, instead of $3.8m$ and a $10\%$ decrease of the $B_C$.

On the other hand, the whole form-finding process can be carried out in this optic. For example, by setting all control point (functions) equal to $z_1 = 5m$, except $D = 2.5m$, the bending coefficient is calculate, $B_C = 0.4$. On the first iteration, $B = z_2$ and the parameter $z_2$ that minimizes the bending behaviour is found for $2.25m$ and $B_C$ = 0.02. On the second iteration $B_C = 0.068$ for $z_3 = 2.55m$, and finally, $B_C$ converges for $z_5 = 3.6m$. Note: As for each iteration, there is only 1 parameter at stake the parametric domains $\Omega_i$ can be set so that the shell´s height respect the boundaries on 5.2. In this case, $E(z_1) = 0.7z_1$.

Table 6.1: Heuristic approach for form-finding

| | Initialization | $1^{st}$ iteration | $2^{nd}$ iteration | $3^{rd}$ iteration |
|---|---|---|---|---|
| $A$ | $z_1 = 5m$ | $z_1 = 5m$ | $z_1 = 5m$ | $z_1 = 5m$ |
| $B$ | $z_1$ | $z_2 = 2.25m$ | $z_2 = 2.25m$ | $z_2 = 2.25m$ |
| $C$ | $z_1$ | $z_1$ | $z_3 = 2.55m$ | $z_3 = 2.55m$ |
| $D = B/2$ | $z_1/2$ | $z_2/2$ | $z_2$ | $z_2/2$ |
| $E$ | $0.7z_1$ | $0.7z_1$ | $0.7z_1$ | $z_5 = 3.6m$ |
| $B_C$ | $0.4$ | $0.02$ | $0.0068$ | $0.0068$ |



(a) $z_1 = 5m$, $z_2 = 5m$, $z_3 = 5m$, $z_4 = 2.5m$ and $z_5 = 3.6m$ $B_C = 0.4$

(b) $z_1 = 5m$, $z_2 = 2.25m$, $z_3 = 5m$, $z_4 = 1.25m$ and $z_5 = 3.6m$ $B_C = 0.02$

(c) $z_1 = 5m$, $z_2 = 5m$, $z_3 = 2.55m$, $z_4 = 2.5m$ and $z_5 = 3.6m$ $B_C = 0.0068$

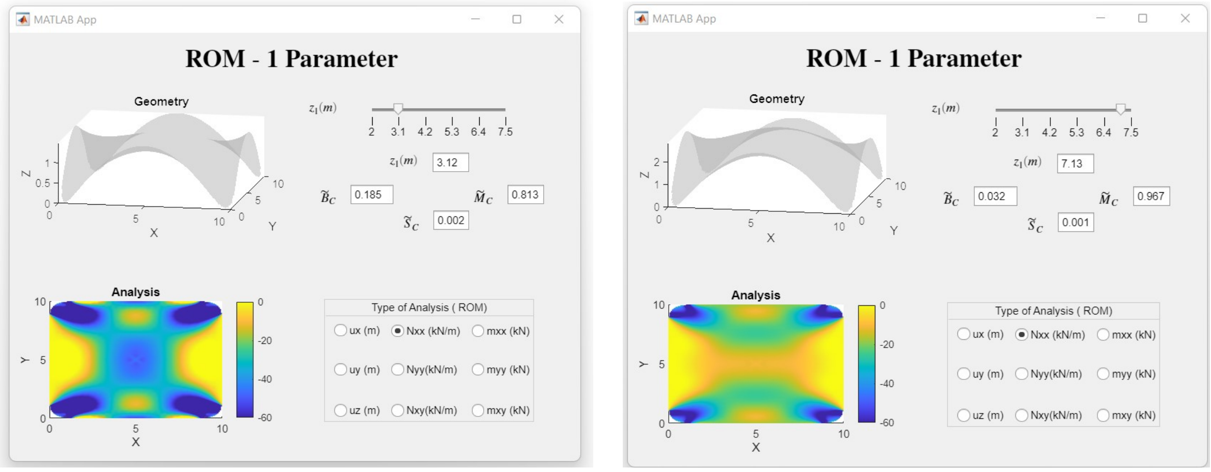
Figure 6.40: Geometries obtained in the iterative process - $z_1$ = 5m

Through the present heuristic process, for an arbitrary central control point, the optimal configuration is easily found. Although this implies neither high memory nor time requirements and provides efficient solutions, it does not allow to have a strong control over the formulation, hence, the generated geometries, which for that reason, are not interesting, from an aesthetic standpoint in contrast with 6.3. Furthermore, it can be used to improve the solution provided by the models as a last iteration.

## 6.6 Visualization of the ROM

A GUI, Graphical User Interface, was developed to solve the problem in visualizing the results, thus displaying information when $D > 2$. The app was programmed in Matlab App Design. For user defined parameters, the app is intended to display the shell´s defined geometry, present the displacements and stresses map, according to the user´s wish and, finally, indicate the structural behaviour of the shell, in terms of energy coefficients. This app was built as an example, therefore, only 1 parameter was used to build the ROM, the process is the same for more parameters. As previously discussed, the ROMs processed carry little to no error. This brings out a huge advantage over the FOM, for a higher number of parameters, in terms of storage needed to obtain similar results. For 1 parameter those advantages are not that obvious, but it was seen for 2 parameters, that the ROM can save up to $95\%$ of memory.

For instance, for $D = 1$, for two different combinations, $z_1 = 3.1m$ and $z_1 = 7.1$ can be promptly assessed. The interface of the app for both inputs is presented in Figure 6.41. As the slider is moved, the structure and its outputs are automatically updated. Simple apps can be developed for more parameters, while consuming little to no memory compared to one produced with the FOM. If an architect expresses its preference for a given type of form, a formulation can be generated to integrate that surface´s type. Then,



ROM - 1 Parameter

$z_1(m)$ 3.12

$\tilde{B}_C$ 0.185 $\tilde{M}_C$ 0.813

$\tilde{S}_C$ 0.002

Type of Analysis ( ROM)

○ ux (m) ● Nxx (kN/m) ○ mxx (kN)

○ uy (m) ○ Nyy(kN/m) ○ myy (kN)

○ uz (m) ○ Nxy(kN/m) ○ mxy (kN)

(a) $z_1 = 3.12m$

ROM - 1 Parameter

$z_1(m)$ 7.13

$\tilde{B}_C$ 0.032 $\tilde{M}_C$ 0.967

$\tilde{S}_C$ 0.001

Type of Analysis ( ROM)

○ ux (m) ● Nxx (kN/m) ○ mxx (kN)

○ uy (m) ○ Nyy(kN/m) ○ myy (kN)

○ uz (m) ○ Nxy(kN/m) ○ mxy (kN)

(b) $z_1 = 7.1m$

Figure 6.41: GUI to showcase results

a straightforward app can be designed to study the effect of changing values of the shell´s parameters, assess its efficiency, and visualizing the stresses and energy coefficients in real time.

# Chapter 7

# Conclusions

## 7.1 Achievements

The present work focused on finding antifunicular forms for geometrically parameterized shells. Proper Generalized Decomposition was used to obtain an explicit parametric solution. Thus, the minimization of the structure´s bending coefficient was easily performed. The ROM relied on an accurate Full Order Model (FOM) which was taken as reference. Hence, a theoretical overview regarding shell history, shell properties, form-finding methods, shell structural analysis, and the PGD was carried out, not only to decide how to easily generate the geometries but also allow a swift interoperability between the different available routines, the FEM and PGD toolbox. This background was key to validate and constantly test the FOM, as the ROM will only be as good as its FOM counterpart.

To stimulate a membrane compressive behaviour, double curvatured, translational, non-developed shells were, essentially, generated through the Bézier Surface formulation, *a priori*. The shells were parameterized for one, two and three parameters taking advantage of the control points from the Bézier surface formulation, defined in Table 7.1.

Table 7.1: Input for the models´s geometry generation

| Model | $1$ Parameter | $2$ Parameters (simple) | $2$ Parameters(extensive) | $3$ Parameters |
|---|---|---|---|---|
| $\Omega$ | $[2;7.5]$ | $[1;8]\times[3;6.5]$ | $[1;8]\times[1;4]$ | $[4,8]\times[2;4]\times[3;5]$ |
| $n_i$ | $60$ | $40\times 20$ | $40\times 20$ | $16\times 8\times 8$ |
| $A$ | $z_1$ | $z_1$ | $z_1$ | $z_1$ |
| $B$ | $5\sin(z_1\frac{\pi}{8})$ | $z_2$ | $5\sin(z_1\frac{\pi}{8})$ | $z_2$ |
| $C$ | $-\frac{z_1}{2}+2.5\sin(z_1\frac{\pi}{8})$ | n/a | $z_3$ | $z_3$ |
| $D$ | $2.5\sin(z_1\frac{\pi}{8})$ | n/a | $2.5\sin(z_1\frac{\pi}{8})$ | $z_2/2$ |
| $E$ | $0.7z_1$ | n/a | $5\sin(z_3\frac{\pi}{6})$ | $(z_1+z_2+2z_3)/4$ |

Through the results compiled in Table 7.2, it is safe to assume that the PGD application to find antifunicular shapes, for geometrically parameterized shells, was successful. The ROMs are not only able to achieve the same solutions as the FOM, but also accurately calculate the displacements, stresses

Table 7.2: ROMs overall performance (AE - absolute error, $n(\mathbf{u})$ number of elements in $\mathbf{u}$)

| Output | $1$ Parameter | $2$ Parameters(simple) | $2$ Parameters(extensive) | $3$ Parameters |
|---|---|---|---|---|
| $M(\tilde{\mathbf{u}})$ | $26$ | $100$ | $100$ | $100$ |
| $\epsilon^*(\tilde{K}_b)$ | $10^{-3}$ | $10^{-2}$ | $5 \times 10^{-2}$ | $10^{-2}$ |
| $dof$ | $9054$ | $9054$ | $9054$ | $9054$ |
| $n(\tilde{\mathbf{u}})/n(\mathbf{u})$ | $30\%$ | $13\%$ | $13\%$ | $10\%$ |
| $AE(\tilde{B}_C^{min})$ | $10^{-5}$ | $0.003$ | $0.0010$ | $0.0002$ |
| $\min(\tilde{\mathbf{B}}_{\mathbf{C}})$ | $0.03$ | $0.04$ | $0.0035$ | $0.0024$ |

and moments for the analysis of each shell, while consuming drastically less memory. An optimization, from the standpoint of the number of modes required to obtain an acceptable approximation was not performed. However, it would decrease the amount of memory needed to store the ROM´s results. The PGD potential to generate high parameterized models relies in creating complex ROMs with little computational memory. Starting from $D = 3$ the results´ visualisation starts to become confusing. It is imperative, to design a GUI, for result post-processing, as in Section 6.6, for the user to obtain the parametric responses in real-time.

The aim of this academic project was to explore all necessary fields of expertise and all the available tools to solve a problem for a few set of parameters, and most importantly to acquire a clear notion on how to enrich the model and how the PGD can be applied to solve similar geometrically parameterized problems.

## 7.2 Future Work

Once again, it is stressed that ROM will only be as good as its FOM counterpart. Constant testing has been performed on the FOM; nevertheless, it would be interesting to conduct some laboratory testing, for a small-scale model of one of the optimal structures found. This experimental analysis would complement the validation process and enrich the FOM.

Some numerical improvements can still be performed. The solution of the PGD system linear solver did not converge for the required thresholds. Improving the mesh, by using less elements, while preventing the shear locking phenomena would boost the effectiveness of this method.

Furthermore, the current objective function only focused on $B_C$, $X(B_C)$. In future developments, by adding the variable $A$ and generating a new objective function, $X(A, B_C)$, that adequately combines these two outputs, would grant even more optimized solutions.

The geometries generated were restricted by the Bézier formulation. Increasing the degree of the polynomials would not be a suitable solution, due to the global control property of the Bézier Surface, thus, for higher degree polynomials, sweeping the value of one parameter will, generally, not lead to significant change of the shell´s form. For this reason, creative free-form geometries can be added to this study. Antifunicular structures are a goal when designing concrete shells, yet, gridshell covers can be built with other materials such as timber and steel. The key ingredient to successfully design shells is the absence of bending. Therefore, implementing numerical form-finding methods (geometrically parameterized methods), such as Force Density Method, Dynamic Relaxation or the Particle Spring System, see [18], in the PGD´s scope would allow to cheaply generate free-form structures. Ultimately, the PGD would be able to handle, as conducted in this work, the structural analysis of each structure, and deliver a light version of the ROM in a GUI.

With all these points in consideration, when this theme is extensively further developed, an interactive and practical tool, capable of easily generating a great variety of forms and delivering an accurate structural analysis, for a certain type of shell cover, may be designed, and employed by architects and engineers.

# Bibliography

[1] A. Jameson, N. A. Pierce, and L. Martinelli. Análisis estructural de algunas obras de félix candela mediante modelos de elementos finitos. In *Hormigón y Acero*, volume 62, pages 61–76. ACHE, 2011.

[2] F. Chinesta, R. Keunings, and A. Leygue. *The Proper Generalized Decomposition for Advanced Numerical Simulations*, pages 1–24. Springer, 2014. ISBN:978-3-319-02864-4.

[3] P. Díez, S. Zlotnik, A. García-González, and A. Huerta. Encapsulated pgd algebraic toolbox operating with high-dimensional data. *Archives of Computational Methods in Engineering (2020)*, 2019. https://doi.org/10.1007/s11831-019-09378-0.

[4] CIMNE. Learning the finite element method with matlab and gid, . http://www.cimne.com/matfem/shells.asp.

[5] P. Díez, S. Zlotnik, A. García-González, and A. Huerta. Algebraic pgd for tensor separation and compression: An algorithmic approach. *Comptes Rendus Mécanique, Volume 346, Issue 7, July 2018, Pages 501-514*. https://doi.org/10.1016/j.crme.2018.04.011.

[6] M.Farshad. *Design and Analysis Shell Structures*. Springer Science+Business Media Dordrecht, 1st edition, 1992. ISBN: 978-94-017-1227-9.

[7] C. Williams. Computational form finding and optimization. In D. V. Sigrid Adriaenssens, Philippe Block and C. Williams, editors, *Shell Structures for Architecture*, pages 21–31. Routledge, 2014. ISBN: 978-0-415-84059-0.

[8] W. Flugge. *Stresses in Shells*. Springer-Verlag Berlin Heidelberg GmbH, 2nd edition, 1973. ISBN: 978-3-642-88293-7.

[9] R. Hooke. *A description of helioscopes, and some other instruments*. Royal Society, 1st edition, 1675. doi: doi.org/10.3931/e-rara-2171.

[10] J. Heyman. *Structural Analysis: A Historical Approach*. Cambridge University Press, 1st edition, 1966. ISBN:0-521-62249-2.

[11] S. Nedev. The catenary - an ancient problem on the computer screen. *European Journal of Physics · August*, 2000. doi: 10.1088/0143-0807/21/5/.

[12] N. M. Patrikalakis and T. M. W. Cho. *Die Graphische Statik*. Meyer & Zeller, 1st edition, 1864.

[13] P. Block, M. DeJong, and J. Ochsendorf. As hangs the flexible line: Equilibrium of masonry arches. *Nexus Network Journal, Volume 8, No. 2, October 200,13-24*, 2006. doi: 10.1007/s00004-006-001.

[14] L. Pelà and P. Roca. Classical approaches and limit analysis part 1: Graphic statics, 2021.

[15] K.-U. Bletzinger and E. Ramm. Computational form finding and optimization. In D. V. Sigrid Adriaenssens, Philippe Block and C. Williams, editors, *Shell Structures for Architecture*, pages 45–55. Routledge, 2014. ISBN: 978-0-415-84059-0.

[16] M. E. M. Garlock and D. P. Billington. Félix candela and heinz isler a comparison of two structural artists. In D. V. Sigrid Adriaenssens, Philippe Block and C. Williams, editors, *Shell Structures for Architecture*, pages 247–257. Routledge, 2014. ISBN: 978-0-415-84059-0.

[17] A. Kilian and J. Ochsendor. Particle-spring systems for structural form-finding. *Journal of the International Association for Shell and Spatial Structures,Volume 46, n. 147, August 2005, Pages 77-84*.

[18] D. Veenendaal and P. Block. Comparison of form-finding methods. In D. V. Sigrid Adriaenssens, Philippe Block and C. Williams, editors, *Shell Structures for Architecture*, pages 115–129. Routledge, 2014. ISBN: 978-0-415-84059-0.

[19] N. Kanta. Design of a thin concrete shell roof. MSc Dissertation. TU Delft. 2015.

[20] F. M. H. de Sena Cardoso. Coberturas em betão armado e pré-esforcado solução estrutural tipo casca. MSc Dissertation. Instituto Superior Técnico. 2008.

[21] Ramaswamy. *Design and Construction of Concrete Shell Roofs*. Krieger Pub Co, 1st edition, 1984. ISBN 0-89874-001-0.

[22] Montoya, P. Jemenez, A. Garcia, and F. M. Cabré. *Hormigón Armado*. Gustavo Gili, 11th edition, 1981. ISBN 8-42520-760-6.

[23] D. P. Billington. *Thin Shell Concrete Structures*. McGraw-Hill, 2nd edition, 1990. ISBN: 0-07-005279-4.

[24] F. Chinesta, R. Keunings, and A. Leygue. *The Proper Generalized Decomposition for Advanced Numerical Simulations*, pages 25–45. Springer, 2014. ISBN:978-3-319-02864-4.

[25] M. Loaiza. A short introduction to Hilbert space theory. *Journal of Physics: Conference Series, Volume 839*, 2017. 10.1088/1742-6596/839/1/012002.

[26] L. Piegl and W. Tiller. *The NURBS book*. Springer, 1 edition, 1995. ISBN:978-3-642-97387-1.

[27] N. M. Patrikalakis and T. M. W. Cho. *Shape Interrogation for Computer Aided Design and Manufacturing*. MIT, 1st edition, 2009.

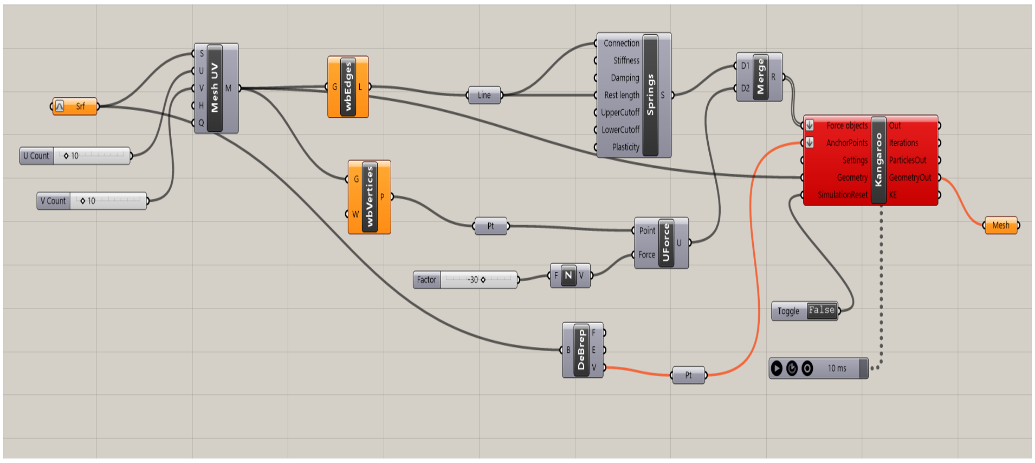
[28] C. Geuzaine and J.-F. Remacle. A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering 79(11), pp. 1309-1331*, 2009.

[29] Grupo de Análise de Estruturas Departamento de Engenharia Civil Instituto Superior Técnico. Análise de Estruturas I: Apontamentos sobre análise de laje. 2018.

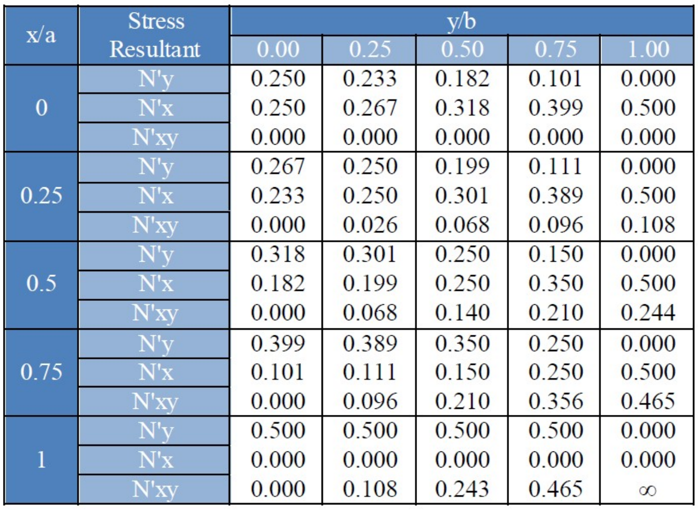
[30] CIMNE. Gid the universal, adaptative and user friendly pre and postprocessing system for computer analysis, . http://www.gidsimulation.com.

# Appendix A

## A.1 Particle Spring method code to generate shell on Grasshopper

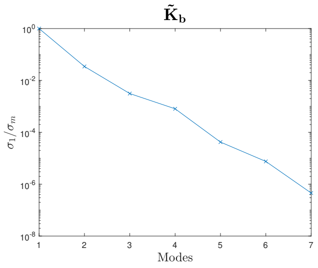


## A.2 Membrane theory elliptic paraboloid stress coefficients c1/c2= 1, as in [23]

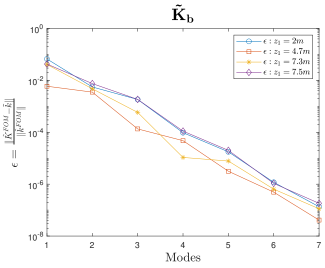


| x/a | Stress Resultant | y/b | | | | |
|---|---|---|---|---|---|---|
| | | 0.00 | 0.25 | 0.50 | 0.75 | 1.00 |
| 0 | N'y | 0.250 | 0.233 | 0.182 | 0.101 | 0.000 |
| | N'x | 0.250 | 0.267 | 0.318 | 0.399 | 0.500 |
| | N'xy | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 0.25 | N'y | 0.267 | 0.250 | 0.199 | 0.111 | 0.000 |
| | N'x | 0.233 | 0.250 | 0.301 | 0.389 | 0.500 |
| | N'xy | 0.000 | 0.026 | 0.068 | 0.096 | 0.108 |
| 0.5 | N'y | 0.318 | 0.301 | 0.250 | 0.150 | 0.000 |
| | N'x | 0.182 | 0.199 | 0.250 | 0.350 | 0.500 |
| | N'xy | 0.000 | 0.068 | 0.140 | 0.210 | 0.244 |
| 0.75 | N'y | 0.399 | 0.389 | 0.350 | 0.250 | 0.000 |
| | N'x | 0.101 | 0.111 | 0.150 | 0.250 | 0.500 |
| | N'xy | 0.000 | 0.096 | 0.210 | 0.356 | 0.465 |
| 1 | N'y | 0.500 | 0.500 | 0.500 | 0.500 | 0.000 |
| | N'x | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| | N'xy | 0.000 | 0.108 | 0.243 | 0.465 | ∞ |

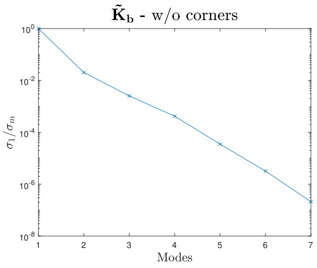
# A.3 Separation of stiffness tensors and force tensor - 1 Parameter



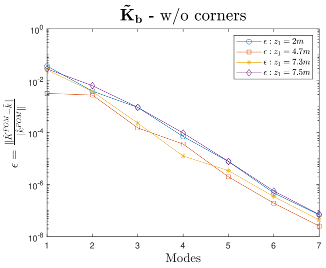
(a)



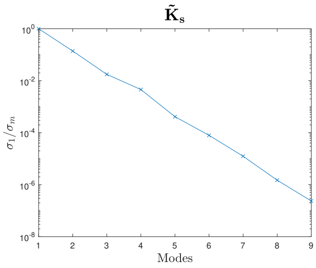
(b)



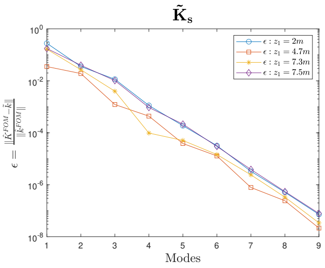
(c)



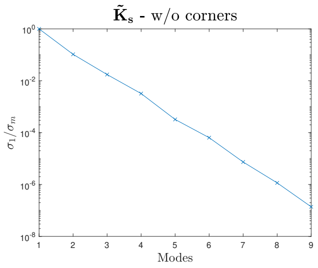
(d)



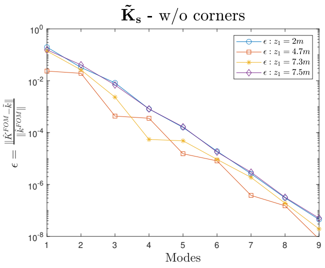
(e)



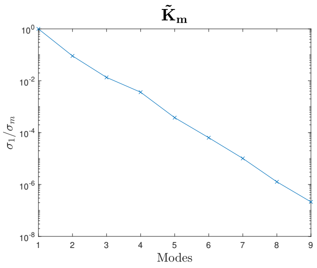
(f)



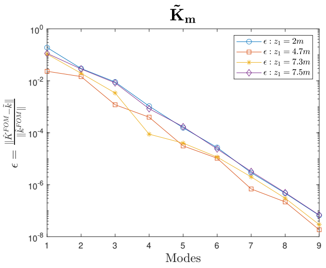
(g)



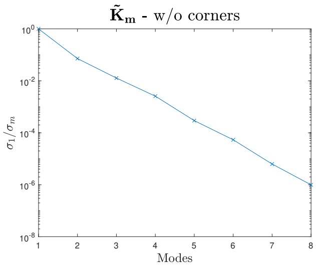
(h)



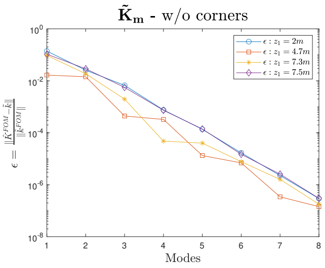
(i)



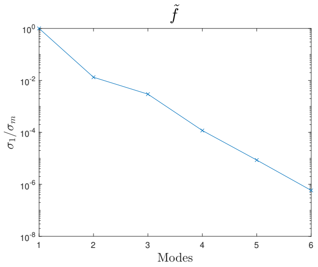
(j)



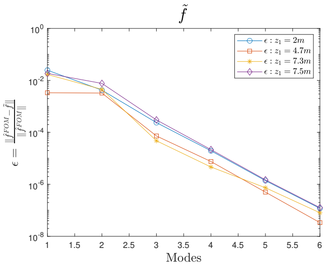
(k)



(l)



(m)



(n)

## A.4 Modal enrichment of approximated displacement tensor - 1 Parameter

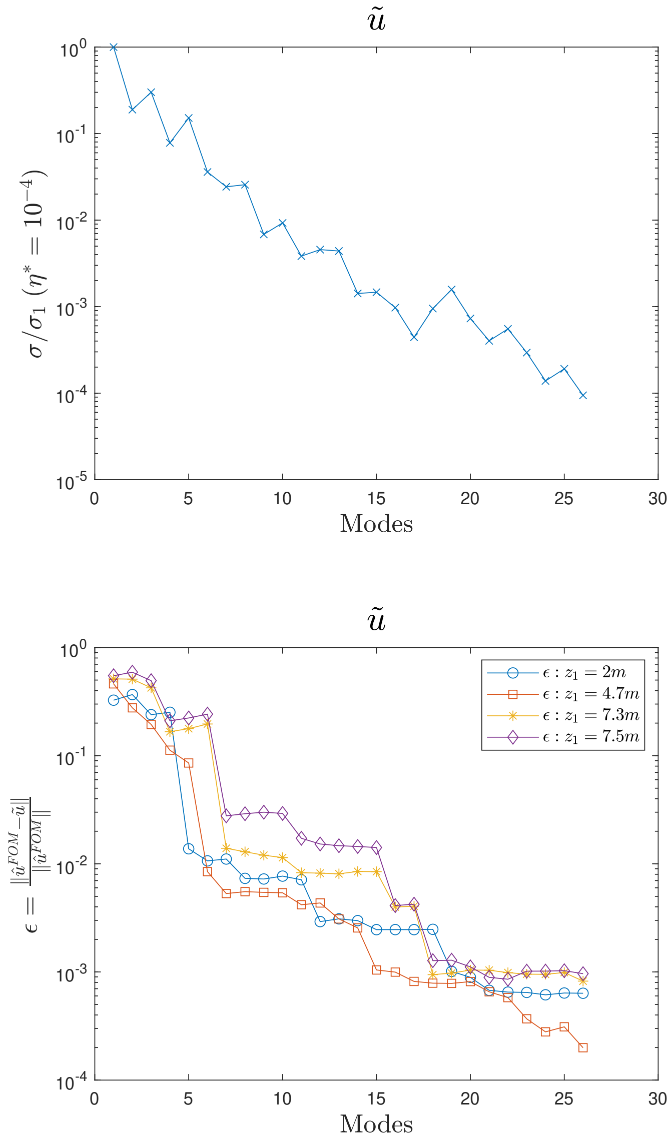
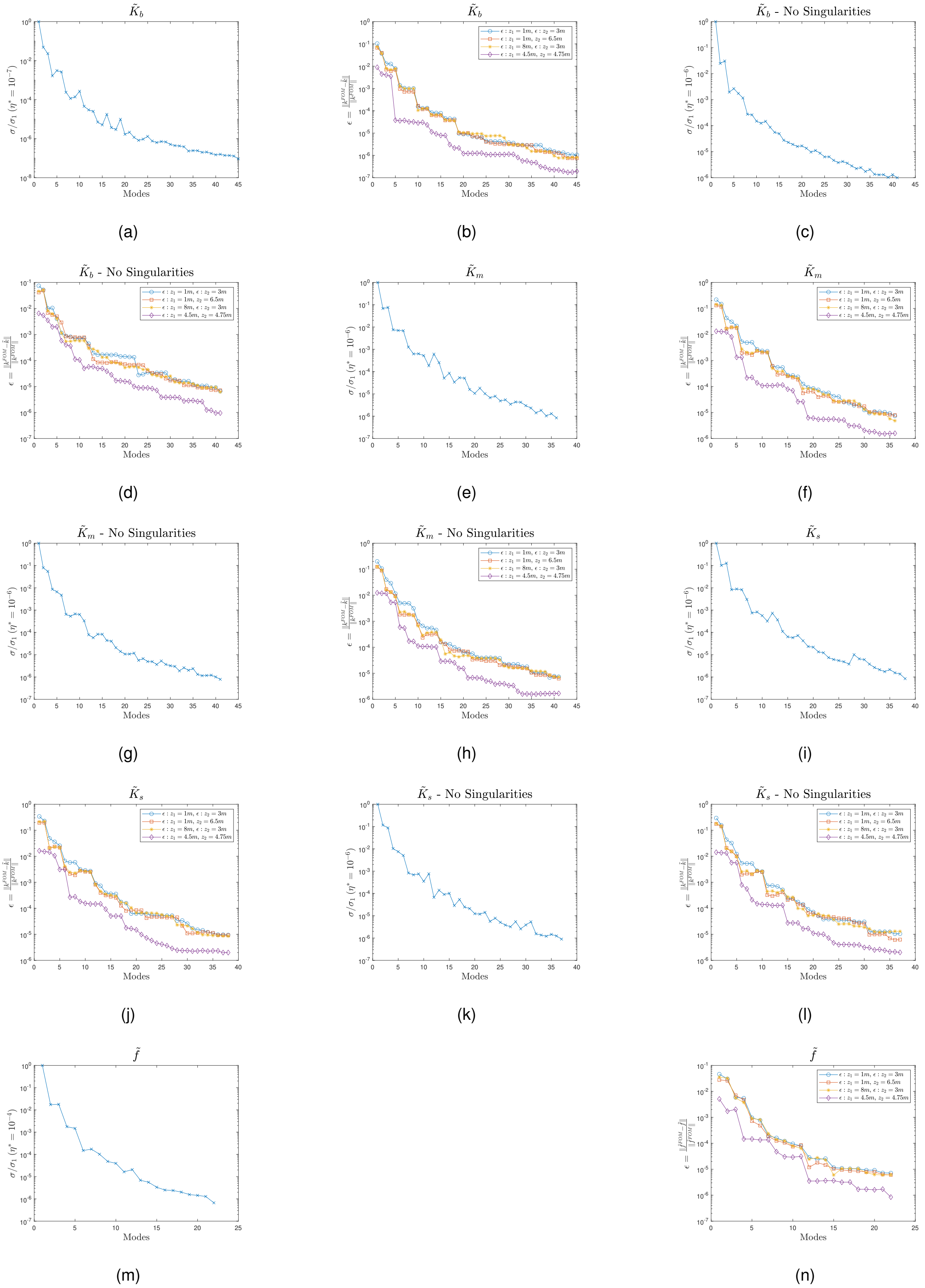


Figure A.1

## A.5 Separation of stiffness tensors and force tensor - 2 Parameters (simple)



(a) (b) (c)

(d) (e) (f)

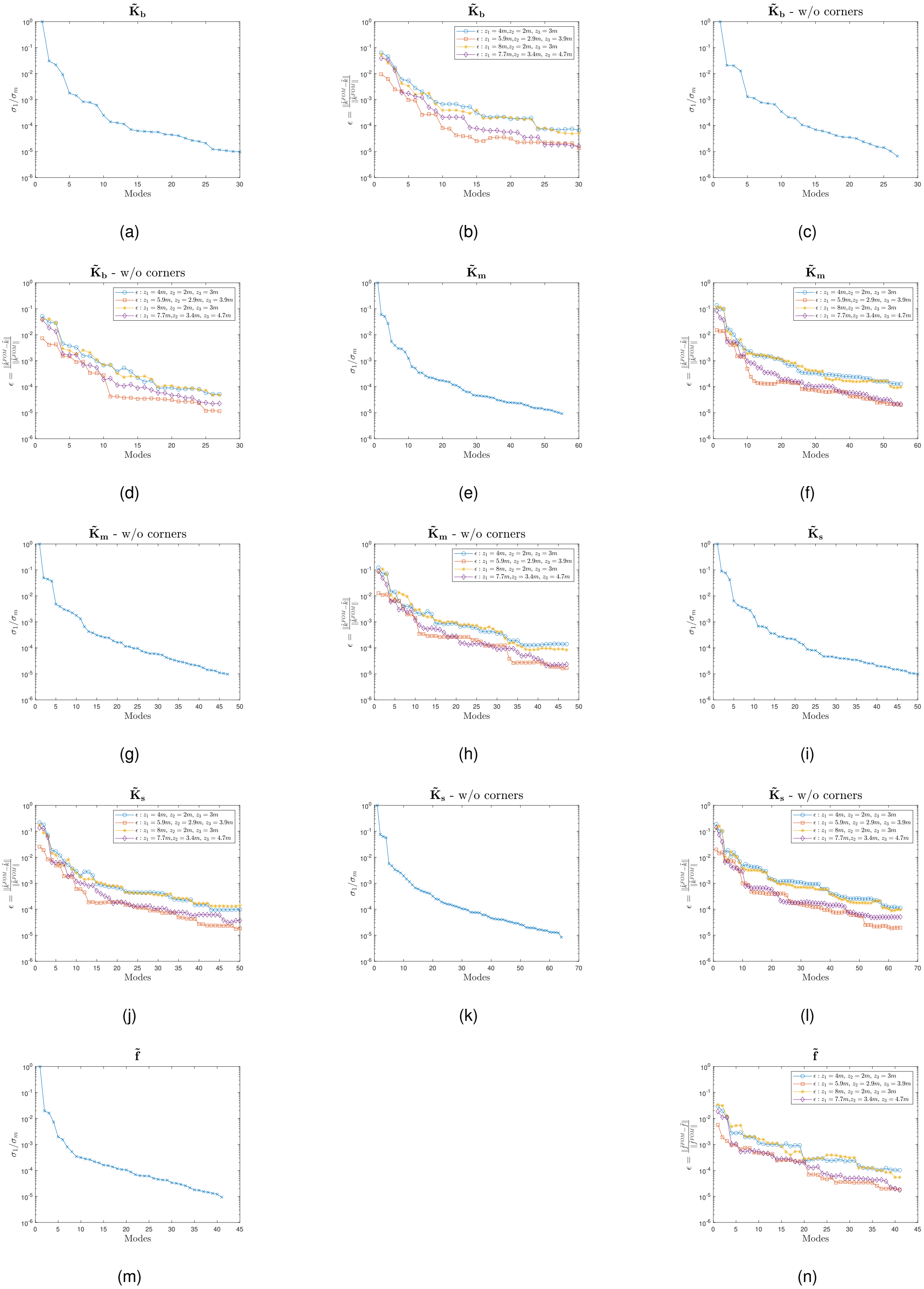
(g) (h) (i)

(j) (k) (l)

(m) (n)

Figure A.2

## A.6 Separation of stiffness tensors and force tensor - 3 Parameters



(a) (b) (c)

(d) (e) (f)

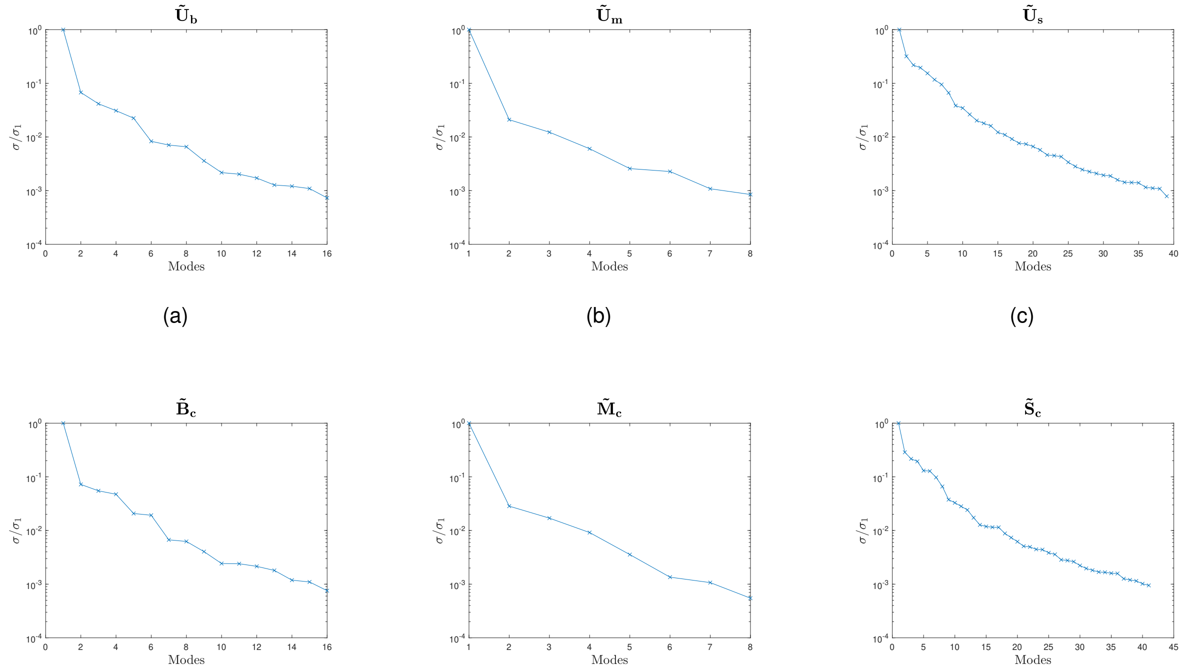
(g) (h) (i)

(j) (k) (l)

(m) (n)

Figure A.3

## A.7 Computation of the energy and energy coefficient tensors - 3 Parameters



(a)

(b)

(c)

(d)

(e)

(f)

Figure A.4