UNIVERSIDADE FEDERAL DA BAHIA

TESE DE DOUTORADO

A Data-Driven Approach for Mass-Spring Model Parametrization Based on Continuous Models

Josildo Pereira da Silva

Programa de Pós-Graduação em Ciência da Computação

Salvador
November-2015

JOSILDO PEREIRA DA SILVA

A DATA-DRIVEN APPROACH FOR MASS-SPRING MODEL PARAMETRIZATION BASED ON CONTINUOUS MODELS

Esta Tese de Doutorado foi apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Doutor em Ciência da Computação.

Orientador: Antônio Lopes Apolinário Jr
Co-orientador: Gilson A. Giraldi

Salvador
November-2015

Ficha catalográfica.

Josildo P.da Silva

A Data-Driven Approach for Mass-Spring Model Parametrization Based on Continuous Models/ Josildo Pereira da Silva– Salvador, November-2015.

93p.: il.

Orientador: Antônio Lopes Apolinário Jr.
Co-orientador: Gilson A. Giraldi.
Tese (Doutorado)– FEDERAL UNIVERSITY OF BAHIA, MATHEMATIC INSTITUTE, November-2015.

TOPICOS PARA FICHA CATALOGRAFICA.
I. Antônio L. Apolinário Jr. II. Gilson A. Giraldi.
III. FEDERAL UNIVERSITY OF BAHIA. MATHEMATIC INSTITUTE.
IV. A Data-Driven Approach for Mass-Spring Model Parametrization Based on Continuous Models.

NUMERO CDD

À Jeina e Malu, filhas amadas que nos reclamos incontidos se tornaram o motivo maior para alcançar a meta que hoje se realiza.

ACKNOWLEDGEMENTS

The work developed throughout the construction process of this thesis is the gathering result of many hands. Collaborated with us the teachers of disciplines, coordinators, secretaries, classmates. That is, a lot of names that in the space provided it would be impossible to thank everyone. So, to all of you my sincere thanks for all the support offered.

However, I would highlight two names: my advisor Antonio Lopes Apolinario Jr, and my co-advisor Gilson Antonio Giraldi. For both goes my most profound gratitude. They not only provided me immense academic support, but they also gave me the valuable suggestions and comments on this thesis.

Eternal gratitude to my family and friends who were extremely understanding, patient and encouraging.

Special thanks also to FAPESP and LNCC by the material support in important moments in research.

No meio do caminho tinha uma pedra...

—CARLOS DRUMMOND DE ANDRADE (Ed.Pindorama, 1930)

RESUMO

Atualmente, a simulação de objetos deformáveis desempenha papel importante em vários campos ligados à Ciência da Computação, como a computação gráfica, projeto assistido por computador, cirurgias assistidas por computador e robótica. Nesse contexto, a simulação de objetos deformáveis com acurácia e em tempo-real é uma tarefa extremamente dificil para as aplicações que requerem simulações mecânicas interativas como são os casos dos ambientes virtuais, simuladores cirúrgicos e jogos. Podemos dividir as abordagens que dão suporte ao tratamento de modelos deformáveis em dois grandes grupos: baseados em mecânica do contínuo, como Método de Elementos Finitos (FEM - *Finite Element Method*) ou Análise Isogeométrica (IGA - *Isogeometric Analysis*); e usando representações discretas, como modelo massa-mola (MSM - *Mass Spring Model*). Métodos baseados na abordagem contínua são conhecidos por seu alto custo computacional e acurácia, enquanto que os métodos discretos, embora simples e adequados para simulações mecânicas interativas, são difíceis de parametrizar. A falta de um método geral baseado em física ou sistemático para determinar a topologia de malha ou os parâmetros do MSM a partir de um material conhecido foi a principal motivação desse trabalho, no sentido de gerar um modelo de baixo custo computacional, como o MSM, a partir de um modelo de alta precisão como o FEM. Portanto, partindo da premissa de simplicidade e adequação do MSM para simulações mecânicas interativas, nesta tese propomos uma metodologia para parametrizar o MSM baseada em modelos contínuos. Desenvolvemos duas abordagens orientadas à dados (*data-driven*) para a parametrização do MSM usando modelos FEM e IGA, este último como referência de derivação com elementos de ordem superior. Com base nos resultados experimentais, a precisão alcançada por estas novas metodologias é mais elevada do que a de outros trabalhos similiares na literatura. Em particular, a nossa proposta alcança excelentes resultados na parametrização do MSM com elementos de ordem superior.

Palavras-chave: Modelos Deformáveis, Modelos Baseados em Física, Parametrização de Modelos Massa Mola, Modelos de Elementos Finitos, Análise Isogeométrica.

ABSTRACT

Nowadays, the simulation of deformable objects play important roles in several fields related to Computer Science such as computer graphics, computer aided design, computer aided surgery and robotics. In this context, simulation of deformable objects with accuracy and in real-time is an extremely difficult task for applications that require interactive mechanical simulations such as virtual environment, surgical simulation and games. We can divide the approaches that offer support for the treatment of deformable models into two groups: based on continuous mechanics, like Finite Element Model (FEM) or Isogeometric Analysis (IGA); and using discrete representations, as Mass-Spring Model (MSM). Continuous-based methods are known for their high computacional cost and accuracy, while discrete methods, although simple and suitable for interactive mechanical simulations are difficult to parametrize. The absence of a general physically based or systematic method to determine the mesh topology or MSM parameters from a known material was the main motivation of this work, in the sense to generate a model of low computational cost, such as MSM, from a model of high accuracy as the FEM. Assuming the premise of simplicity and suitability of the MSM for interactive mechanical simulations, in this thesis we propose a methodology to parametrize the MSM based on continuous models. We developed two data-driven approaches to the parametrization of the MSM by using FEM and IGA models, the latter as reference for derivation with higher order elements. Based on experimental results, the precision achieved by these new methodologies is higher than other similar approaches in literature. In particular, our proposal achieves excellent results in the parametrization of the MSM with higher order elements.

Keywords: Deformable Models, Physics Based Models, Mass-Spring Parameterization, Finite Element Model, Isogeometric Analysis.

CONTENTS

Chapter 1—Introduction 1

1.1 Motivation . . . 1
1.2 Context . . . 4
1.3 Objectives and Contributions . . . 4
1.4 Thesis Outline . . . 5

Chapter 2—Continuous Models 7

2.1 Continuum Elasticity Theory . . . 9
2.2 Finite Element Method . . . 10
2.3 Basic B-Spline Theory . . . 14
2.3.1 Multivariate B-Spline Basis Functions . . . 16
2.3.2 B-Spline Geometries . . . 17
2.4 Non-uniform Rational B-Spline . . . 18
2.4.1 Rational B-Spline Curves . . . 18
2.4.2 B-Spline Rational Surfaces and Solids . . . 19
2.5 Isogeometric Analysis . . . 20
2.5.1 Element Formulation . . . 22
2.5.2 Discretization and Analysis . . . 24
2.6 Conclusions . . . 27

Chapter 3—Discrete Models 29

3.1 Mass Spring Model . . . 30
3.2 Parameters Identification . . . 32
3.2.1 Nodes Mass Identification . . . 34
3.2.2 Springs Stiffness Identification . . . 34
3.2.3 Simulated Annealing . . . 36
3.3 Conclusions . . . 37

Chapter 4—Parameterization Based on FEM 39

4.1 Acceleration Cost Function . . . 41
4.2 Grouping Spring . . . 42
4.3 Deformation Measurement . . . 44
4.4 Proposed Method . . . 44
4.5 Results . . . 46
4.6 Discussions . . . 55

Chapter 5—Parameterization Based on IGA 57

5.1 The NURBS Mass Spring Model 58
5.2 NMSM Parameterization by IGA 60
5.3 Results 64
5.4 Discussions 74

Chapter 6—Conclusions and Perspectives 75

6.1 Future Works 77

Appendix A—Multi-model Analysis Framework 87

LIST OF FIGURES

1.1 The simulation based on continuum mechanics. 3

2.1 Displacement field in continuous body. 9
2.2 Finite element discrezatization. 11
2.3 The solution of continuum problem by FEM. 13
2.4 The solution of continuum problem by IGA. 21
2.5 The IGA meshing. 23
2.6 The geometry mapping in the IGA. 25
2.7 The IGA spaces. 26

4.1 The mapping FEM to MSM. 45
4.2 The MSM mesh for square elastic membrane. 48
4.3 The time integration analysis. 49
4.4 The elastic membrane mesh. 50
4.5 The time evolution of deformation. 51
4.6 The FEM meshes. 52
4.7 New discretization for square membrane. 54

5.1 The mapping IGA to NMSM mesh. 60
5.2 Overview of mapping from the NURBS-based IGA model to NMSM. . . 62
5.3 The control and physical meshes for 2D NURBS elastic objects. 65
5.4 The control and physical meshes for 3D NURBS elastic objects. 66
5.5 Error representation for elliptical membrane. 70
5.6 Error representation for coupled spheroid. 71
5.7 Comparative deformation perceptual in time. 73

A.1 The packages diagram in UML for MAF. 88
A.2 The UML class diagram to the packages in the MAF. 90
A.3 The UML class diagram to the package *Element* in the MAF. 91
A.4 The UML class diagram to the package *ElementGeometry* in the MAF. . 91
A.5 The UML class diagram to the packages *ElementDriver* and *ShapeFunction* in the MAF. 92
A.6 The *makSetting* class with its attibutes and methods. 92

LIST OF TABLES

4.1 The experiments overview. 47
4.2 MSM parameters derived by acceleration approach. 53
4.3 Comparative results with matrix approach. 53
4.4 Stiffness coefficients and errors for different Poisson's ratio. 54

5.1 The geometry data for elastic objects in experiments. 64
5.2 Number of groups and spring coefficients 67
5.3 Quantitative results 2D/3D objects 67
5.4 Sensitivity to mesh resolution mesh 68
5.5 Grouping spring for the 2D geometry experiments 69
5.6 Comparative quantitative results 70
5.7 Comparative under linear basis functions 72

LIST OF ABBREVIATIONS

CAD - Computer Aided Design

CAGD - Computer Aided Geometric Design

D-NURBS - Dynamic NURBS

FEM - Finite Element Model

IGA - Isogeometric Analysis

MAF - Multi-model Analysis Framework

MSM - Mass Spring Model

NMSM - NURBS Mass Spring Model

NURBS - Non-uniform Rational B-Spline

PDE - Partial Differential Equation

Chapter 1

INTRODUCTION

Deformable objects span a wide range of different object types with different physical properties, such as jelly, cloth, biological tissue, water, wind. The behavior of deformable objects is considered more difficult to deal with than rigid body simulation for several reasons. First, degrees of freedom needed to describe the motion of a single deformable object can boost in large scale compared to a single rigid object (REDDY, 2013). Second, since deformable objects could potentially self-intersect, geometric problems such as detecting collisions and self-intersections should be handled. Third, the physics of deformable objects is governed by a constitutive law describing relationship between physical properties inside object, such as stress and strain (ERLEBEN et al., 2005). So, the behavior simulation of a deformable model in an interactive system with precision and in real time is an extremely difficult task.

1.1 MOTIVATION

Nowdays, the behavior simulation of deformable objects play important roles in several fields such as computer graphics, computer aided design, computer aided surgery, robotics, games, and other simulation applications (WU; WESTERMANN; DICK, 2015; ROYER et al., 2015). In computer graphics applications, deformable models have been widely used for the animation of clothing (PROVOT, 1996; LONG; BURNS; YANG, 2011), facial expression and characters (LEE; TERZOPOULOS; WATERS, 1995; KäHLER; HABER; SEIDEL, 2001). In computer aided design, deformable models are used as important tool for geometric modeling which enables the modeler sculpting complex shapes in a easy fashion (TERZOPOULOS; QIN, 1994; IX; QIN; KAUFMAN, 2001). In computer aided surgery, deformable models are needed in simulation of human tissues which demand both real time and physically realistic modeling of complex, non-linear, deformable tissues (XU et al., 2011; COMAS, 2010). In robotic, they have been considered in the path planning process as a relevant component on perceiving the surroundings and modeling the environment (FRANK et al., 2014). In computer games, non-physically and physically based techniques have been widely used to improve the game engine performance. Here, the choice depends on the dilemma between real-time and accuracy. All aforementioned fields share a common goal: they search to find out the visually and physically plausible response which has caused the growth of works to the area referred as physics-based deformable model (NEALEN et al., 2006).

The physics-based deformable model is an interdisciplinary field that elegantly combines continuum mechanics, numerical analysis, differential geometry, approximation theory and computer graphics into a powerful toolkit. Hence, deformable models based on physical laws have been developed in an attempt to exhibit a realistic behavior with the desired accuracy. Among the models that exhibit a realistic behavior there are two categories : those based on continuum mechanics and those using discrete representations. The first class is usually generated by finite element models (FEM) and among the discrete models, the most commonly used is the mass-spring model (MSM).

The FEM is a common and accurate way to compute complex deformations of deformable object due to its connection with continuum mechanics. In this context, constitutive laws are used for the computation of the symmetric internal stress tensor and a conservation law gives the final partial differential equation (PDE) that governs the dynamics of the material (ERLEBEN et al., 2005; NEALEN et al., 2006). The FEM performs discretization of continuous object into a set of elements such that global deformation is considered locally in each element. Deformation within an element is taken into account by the interpolation of nodal displacements as governed by the underlying constitutive laws (CHOI et al., 2004).

In Figure 1.1 one can see outlined the pieces utilized in the simulation of deformable object based on continuum mechanics. In summary, the continuum mechanic gives the formulation of partial differential equation to the deformable object which is solved using the finite element method and numerical time integration. A visual simulation successively returns the dynamic solution with the deformed geometry and forces, which then can be rendered on displays.

The FEM has an alternative methodology that was pointed out in (HUGHES; COTTRELL; BAZILEVS, 2005) considering a NURBS-based framework, which nowadays is known as Isogeometric Analysis (IGA) (COTTRELL; HUGHES; BAZILEVS, 2009). Isogeometric analysis is a technique for the discretization of partial differential equations, inspired by the desire to unify the fields of computer aided geometric design (CAGD) and the finite element method (FEM). The IGA uses the same kind of basis functions that CAGD uses, such as B-splines, NURBS, T-splines, subdivision surfaces, etc., instead of traditional C^0-continuous Lagrange finite element interpolatory polynomials (TEMIZER; WRIGGERS; HUGHES, 2011). So, one of the advantages of IGA models is the usage of a common geometric representation for creating CAD models, meshing and numerical simulation.

The MSM has been a discrete approach widely employed for simulation of deformable object. The MSM approximates a continuous body by a finite set of mass points (nodes) and massless springs connecting them; therefore, it is also called a particle system approach. The deformation of the body is described as a displacement of mass points, evoked by internal and external forces. MSM are simple to implement and can be faster

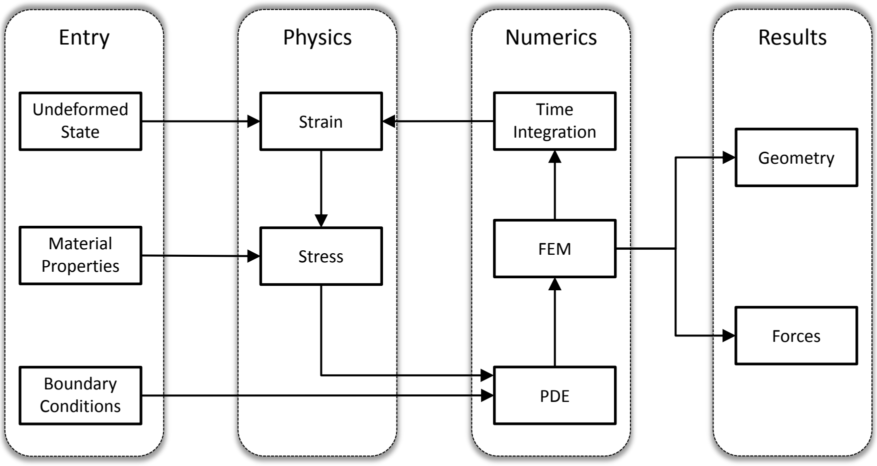


Figure 1.1: Outline of the simulation of deformable object based on continuum mechanics

then the continuous ones, and so, more suitable for real time and interactive applications (NEALEN et al., 2006). There are three important advantages that make the MSM very atractive: simplicity in its mathematical formulation, great versatility for topological changes, and a well-suited data structure for parallel computing.

However, the main limitation of the MSM is the difficulty of designing them to represent the mechanical behavior of deformable objects with enough accuracy (LLOYD; SZéKELY; HARDERS, 2007). In practice, to design the MSM with sufficient accuracy we need a proper selection of mesh topology, spring and damping constants, and nodal mass values. The typical practice in the literature is somewhat *ad hoc*, the cell types and connectivities are empirically assumed, usually based on the structure of the geometric model at hand, and the parameters are either hand tuned to get a reasonable behavior (NATSUPAKPONG; ÇAVUSOGLU, 2010). In this direction, an interesting strategy could be to use the measured elasticity to set the parameters in the MSM leading us to find out a method which is able to establish a link between the mass spring model and the constitutive equations.

Many improvements in the MSM designing have been presented by several authors in order to become the simulation of deformable models accurate and realistic. Some of them have conducted research that aim to developing new methods for obtaining of spring coefficients (LLOYD; SZéKELY; HARDERS, 2007; SILVA; GIRALDI; APOLINARIO, 2015) whereas others have proposed modify the traditional structure of the MSM by including force-based constraints (CHOI et al., 2004; BAUDET et al., 2009) or by adding special hinge as angular or bending springs (PROVOT, 1996; BOURGUIGNON; CANI, 2000; GIRALDI; ORTIZ; JR, 2006). Among them there is a consensus that exist an acceptable approach to computing the nodal masses values as in (DEUSSEN; KOBBELT; TüCKE, 1995; BAUDET, 2006), however, the problem of selecting the appropriate spring and damping constants is still unsolved (NATSUPARKPONG, 2009; SAN-VICENTE; AGUINAGA; CELIGUETA, 2012).

1.2 CONTEXT

In computer graphics applications and related fields, the search for models that enable real-time dynamic simulation of deformable objects is an active area of research (LEE et al., 2012; LIU et al., 2013; WU; WESTERMANN; DICK, 2015). In these fields, a wide variety of physically based models has been developed to address the challenge of simulating natural elements and deformable materials (MOORE; MOLLOY, 2007; JACOBSON et al., 2014).

Notwithstanding of the MSM advantages, it is known that methods that are based on the continuum mechanics are, in general, more realistic than their MSM counterparts (GELDER, 1998; LLOYD; SZéKELY; HARDERS, 2007; DELINGETTE, 2008). In fact, the mechanical systems in elasticity theory are characterized by their macroscopic parameters (Young's module and Poisson's ratio) and constitutive equations (ZIENKIEWICZ; TAYLOR; ZHU, 2005). However, there is no general physically based or systematic method in the literature to determine the mesh topology or MSM parameters from known such set up (SAN-VICENTE; AGUINAGA; CELIGUETA, 2012).

When comparing mass-spring and continuous models the following questions arise: (1) Which set of elastic material properties can be accurately simulated by a particular spring mesh model ? (2) How to derive a mass-spring system from continuum mechanics ? (3) How to specify system parameters (masses, spring constants, mesh topology) in order to match physical requirements ?

The questions (1) and (3) have been more or less addressed by works that adapt MSM to describe different elastic behaviors such as anisotropy, heterogeneity, non linearity and also incompressibility (see (BAUDET et al., 2009) and references therein).

In this work, we are interested to answer question (2) focusing on deformable objects in interactive mechanical simulations. Deriving the MSM parameters of a continuous models allow us to emulate the mechanical behavior of non-rigid objects due to external influences in real-time with physically realistic deformation. So, a new method to determine MSM parameters, such as mass, spring stiffness coefficient and damping constants, using continuous models as reference will be presented.

Besides, and specially important for this work, the MSM derivation from FEM model were restricted to linear elements (NATSUPAKPONG; ÇAVUSOGLU, 2010; SAN-VICENTE; AGUINAGA; CELIGUETA, 2012). Up to the best of our knowledge, FEM models with higher order elements have not been considered in the MSM derivation literature. On the other hand, the isogeometric analysis framework, allows to use NURBS to represent both geometry and physical fields in the solution of problems governed by partial differential equations (PDE) and also enable higher-order elements.

1.3 OBJECTIVES AND CONTRIBUTIONS

The main goal of this thesis is to propose computational methods to the MSM parameterization with application in interactive mechanical simulations of deformable objects using continuous models as reference of derivation in order to ensure maximum accuracy. To achieve this main goal we propose two approaches to determine the MSM parameters : one using the FEM as reference model; and other one with NURBS-based IGA as reference. In both the aim is to determine the MSM parameters (mass, spring stiffness coefficient and damping constants) in a systematic procedure that allows both the models behave similarly.

Thereby, this thesis has important contributions to the parameterization of mass-spring models that is a topic of great interest to physics-based deformable models area. The main contribution is the proposal of the two new approaches to design mass-spring models, that is, to define their inner parameters (mass, stiffness coefficient and damping constant) and mesh topology in order to represent the mechanical behavior of deformable objects with enough accuracy. As secondary but no less important contributions, we highlight: an innovative proposal to derive the MSM parameters by using isogeometric analysis; a proposal of two new objective functions for data-driven methods; an innovative application of the higher order elements for the MSM parameterization; an improvement into accuracy of geometry description applicable to interactive mechanic simulations; an effective heuristic to deal with tradeoff between geometry complexity and error rate in optimization problems involving springs parameters (stiffness coefficient, rest length); a software structure named Multi-model Analysis Framework (MAF) that was used to developed all our work.

1.4 THESIS OUTLINE

The remaining content of this thesis is organized as follows: in Chapter 2, we present the theoretical basis for two continuous models (FEM and IGA) that will be used in the proposals to the MSM parameterization. In this chapter, we highlight the relevant characteristics to emulate the mechanical behaviour of deformable objects accurately in addition to describing briefly the B-Spline and NURBS theories that is a prerequisite for isogeometric analysis. The description to the parameterization strategies together with a simplified formulation of the MSM are presented in the Chapter 3. The aim of this chapter is to show how it is difficult the MSM parameterization and how different categories of approaches face this issue. The proposal of parameterization based on acceleration is carried out in Chapter 4. Here, we use the FEM as reference model to compute the MSM parameters in a data-driven strategy. In Chapter 5 we propose a mass-spring model NURBS-based wherein control points are treated like mass points connected by massless springs and the parameterization is accomplished by using the NURBS-based IGA as reference model. Our discussion of results and contributions of this thesis are presented in Chapter 6. Finalizing the content, we have the Appendix A where we briefly present our software structure named Multi-model Analysis Framework(MAF) that was used to develop all our work.

Chapter 2

The applications with interactive mechanical simulations include mainly animation, virtual environments, surgical training and games. In these applications it is required to emulate the mechanical behaviour of non-rigid objects (deformable objects) in real-time. The mechanical behavior of deformable objects can be described by continuum elasticity theories adding accuracy to simulations. The finite element method (FEM) is a continuous deformable model often used to solve elasticity problems. Recently isogeometric analysis (IGA) has been applied as an alternative methodology to the FEM with promising results. In this chapter we present the theoretical basis for these two models highlighting the relevant characteristics to emulate the mechanical behaviour of deformable objects accurately.

CONTINUOUS MODELS

The simulation of deformable objects has been the focus of attention in several fields such as computer graphics, computer aided design, robotics, computer aided surgery, games, and other simulation environments. All aforementioned fields share a common goal: they search to find out the visually and physically plausible deformation in a computer-aided enviroment. This search has caused the growth of works to the area referred as physics-based deformable model (NEALEN et al., 2006; LONG; BURNS; YANG, 2011; WU; WESTERMANN; DICK, 2015; ROYER et al., 2015).

For the computer science community, the genesis of physics-based deformable models is a seminal manuscript on elastically deformable models where Terzopoulos (TERZOPOULOS et al., 1987) added physical properties directly in a graphical object and inaugurated the term *deformable models*. According to Meier *et al* (MEIER et al., 2005), a deformable model can be defined in either one dimension (lines and curves), two dimensions (surfaces), or three dimensions (solid objects). As noted by these authors, the interactive mechanical simulation is an important field of application for deformable models. In this area is required of the deformable model emulate the mechanical behaviour of non-rigid objects due to external influences in real-time with physically realistic deformation.

These real time applications (that require interactive mechanical simulations) include mainly animation, virtual environments, surgical training and games. In animation, the simulation must be fast enough to provide useful feedback to the animator. In virtual environments, the objects must deform in real time, in response to user input. In surgical training and games with haptic force feedback, forces due to the deformation must also be computed in real time (JAMES; PAI, 1999; ZERBATO; GALVAN; FIORINI, 2007; HUANGFU; YAN; LIU, 2013). The primary issue to be solved by these applications is how to deal with the tradeoff between interactivity and accuracy.

The techniques for simulating deformable objects can be grouped into two categories (MORRIS; SALISBURY, 2007): constitutive and non-constitutive models. That for convenience, we will name the constitutive and non-constitutive models as the continuous and discrete models, respectively. Approaches based on continuous models generally use equations from physics to describe how a material will behave in terms of physical constants that describe real materials (Poisson's ratio, Young's modulus). These methods are traditionally accurate relative to discrete methods. However, the continuous methods are generally associated with significant computational overhead, often requiring solutions to large linear systems, and thus cannot generally be applied to interactive simulations. When these approaches are adapted to interactive simulations, generally through an assumption of linear elasticity.

The discrete approaches rather than using physical constants (e.g. Young's modulus) to describe a material, such approaches describe objects in terms of constants that are particular to the simulation technique employed. Many approaches in this category are variants on the grid of masses and springs, whose behavior is governed by spring constants that can't be directly determined for real materials. In general, these methods are thus not accurate in an absolute sense. However, many approaches in this category are proper for interactive simulations.

The relationship between continuous and discrete models also can be explored in order to take advantage of vocation of discrete models for interactive mechanical simulations. In general, the researchers try to find out ways to design discrete models so that continuous and discrete models behave similarly (BIANCHI et al., 2004; BAUDET et al., 2009). An approach often used to perform this task is to derive the discrete model parameters, in particular the spring coefficients, from the continuous models (LLOYD; SZéKELY; HARDERS, 2007; NATSUPAKPONG; ÇAVUSOGLU, 2010; SAN-VICENTE; AGUINAGA; CELIGUETA, 2012; SILVA; GIRALDI; APOLINARIO, 2015).

Therefore, in order to take advantage of vocation of discrete models for interactive mechanical simulations our research is directed to find out a set of parameters for the discrete model from a continuous model so that both behave in similarly. In this sense, the knowledge about theoretical basis of the continuous models has a paramount importance.

The finite element method (FEM) is a continuous deformable model often used to solve elasticity problems(ZIENKIEWICZ; TAYLOR; ZHU, 2005; RAO, 2004; SADD, 2009). Recently isogeometric analysis (IGA) has been applied as an alternative methodology to the finite element methods with promising results (ESPATH; BRAUN; AWRUCH, 2011; COTTRELL; HUGHES; BAZILEVS, 2009).

In this chapter we present the theoretical basis for these two continuous models. In particular, we will bring a more detailed treatment of the finite element theory to be able to exploit it opportunely in designing of the discrete model and software implementation.

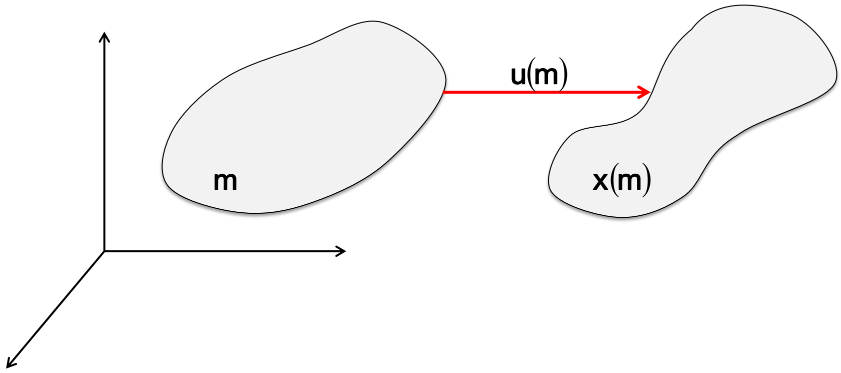


Figure 2.1: Displacement field in continuous body.

2.1 CONTINUUM ELASTICITY THEORY

Continuum mechanics is a branch of mechanics that deals with the analysis of the kinematics and mechanical behavior of rigid and deformable bodies. It is concerned with the mechanical behaviour of solids and fluids on the macroscopic scale ignoring the discrete nature of matter, utilizing the average values of physical quantities such as density, displacement, velocity and so on as continuous (or at least piecewise continuous) (SPENCER, 2004; REDDY, 2013).

The theory of elasticity is a sub-branch of continuum mechanics that deals with the mechanical behaviour of deformable solid bodies. Following theoretical development made by Nealen *et al* (NEALEN et al., 2006), in continuum mechanic a deformable body is typically represented by its undeformed shape (also called equilibrium configuration, rest or initial shape) and its material properties that define how it deforms under applied forces. Assuming the rest shape as a continuous connected region $M \subset \mathbb{R}^3$, thus the coordinates $\mathbf{m} \in M$ of a point in the object are called material coordinates of that point. Considering a discrete case of M, we have a discrete set of points that represent the rest shape of the object. Under the action of forces, the object deforms and a point originally at location $\mathbf{m}$ (i.e with material coordinates $\mathbf{m}$) moves to a new position $\mathbf{x}(\mathbf{m})$, that correspond to the spatial coordinates of that point. Once new positions are defined for all material coordinates $\mathbf{m}$, $\mathbf{x}$ is a vector field defined on M. On the other hand, the deformation can also be specified by the displacement vector field $\mathbf{u}(\mathbf{m}) = \mathbf{x}(\mathbf{m}) - \mathbf{m}$ defined on M, as we can observe in Figure 2.1. Thus, we can compute the elastic strain ϵ which is dimensionless quantify that in the 1D case can be expressed simply by $\frac{\triangle l}{l}$, where l is the length. Spatially constant displacement field represents a translation of the object with absence of strain. Therefore, it becomes clear that strain must be measured in terms of spatial variations of the displacement field $\mathbf{u} = \mathbf{u}(\mathbf{m}) = (u, v, w)^T$.

In general, computer graphics and related fields have chosen to express strain by

$$\epsilon_G = \frac{1}{2}\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T + (\nabla \mathbf{u})^T \nabla \mathbf{u}\right), \tag{2.1}$$

$$\epsilon_C = \frac{1}{2}\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right), \tag{2.2}$$

where $\epsilon_G \in \mathbb{R}^{3\times 3}$ is Green's non-linear strain tensor,$\epsilon_C \in \mathbb{R}^{3\times 3}$ is Cauchy's linear strain tensor, and the gradient of the displacement field is denoted by the 3 by 3 matrix $\nabla \mathbf{u}$.

The computation of the stress σ takes place through a constitutive law (or also called material law). Usually, the symmetric internal stress tensor $\sigma \in \mathbb{R}^{3\times 3}$ for each material point $\mathbf{m}$ based on the strain ϵ at that point is computed using Hooke's linear material law given by

$$\sigma = \mathbf{E}\epsilon, \tag{2.3}$$

where $\mathbf{E}$ is fourth-order tensor which relates the coefficients of the stress tensor linearly to the coefficients of the strain tensor.

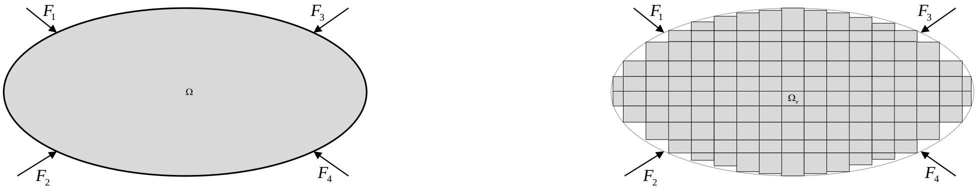
For isotropic materials (a material which has the same mechanical properties in all directions), the coefficients of $\mathbf{E}$ depend only on Young's modulus and Poisson's ratio. Two very common elastic models used in deformable bodies simulations are: a) the linear elastic model, using the linear Cauchy strain tensor ϵ_C with Hooke's linear material law; b) The St. Venant-Kirchoff elastic model using the non-linear Green's strain tensor ϵ_G in combination with Hooke's linear material law.

2.2 FINITE ELEMENT METHOD

The finite element method (FEM) is a systematic technique to obtain numerical solution of partial differential equations (PDEs) describing the continuum behavior of deformable objects. In other words, the objective of FEM when applied to deformable object simulation is to solve PDEs provided by the elasticity theory (for more details about elasticity theory the reader is referred to authors (ZIENKIEWICZ; TAYLOR; ZHU, 2005; SADD, 2009)).

In the finite element method, the actual continuum or body of matter, such as a solid, liquid, or gas, is represented as an assemblage of subdivisions called finite elements. These elements are considered to be interconnected at specified joints called nodes or nodal points (LOGAN, 2011). Hence this process of modeling a continuous body by dividing it into an equivalent system of smaller bodies or units (finite elements) interconnected at points common to two or more elements (nodal points or nodes) and/or boundary lines and/or surfaces is called *discretization (RAO, 2004)* . In Figure 2.2 it is shown a 2D elliptical continuous body (a) with its discretization (b) under loads $F_1, \ldots F_4$.

Since the actual variation of the displacement inside the continuum is not known and it is necessary to calculate the deformation, one can assume that the variation of the displacement inside a finite element can be approximated by a simple function. These approximating functions (also called interpolation models) are defined in terms of the values of the displacements at the nodes. Thus, each finite element has associated some



(a)

(b)

Figure 2.2: Finite element discrezatization. The continuous body (a) and its discretization(b).

interpolation function that define the displacement of any internal point in function of the displacement of the nodes of that finite element. In this way, solving the mechanical problem just in the nodes of the finite elements gives the approximated displacement field of the whole continuos body.

Let express the previous ideas as mathematical fashion. Thus, let (x_1, x_2, x_3) be an inertial cartesian frame of reference in $\mathbb{R}^3$. The key idea in FEM is to subdivide the object domain Ω into subdomains or finite elements Ω_e, $e = 1, 2, \cdots, n_e$, and consider the following finite element approximation on each element

$$\mathbf{x}\left(x_1, x_2, x_3, t\right) = \sum_{i=1}^{n} \mathbf{N}_i^e\left(x_1, x_2, x_3\right) \mathbf{x}_i^e\left(t\right), \tag{2.4}$$

where $\mathbf{x}$ is the deformation field of the body as a function of the material point coordinates (x_1, x_2, x_3) and time t, $\mathbf{N}_i^e$, $i = 1, 2, \cdots, n$ are the interpolation functions, and $\mathbf{x}_i^e \in \mathbb{R}^3$, $i = 1, 2, \cdots, n$ are the coordinates of the nodes of the element Ω_e.

In what follows we suppose that there is a reference configuration for the body, denoted by Ω_0, with mass density and external body force field given by ρ_0 and $\mathbf{b}_0$, respectively. At the element level without damping, the FEM equations have the form (NATSUPAKPONG; ÇAVUSOGLU, 2010):

$$\mathbf{M}^e \ddot{\widehat{\mathbf{x}}}^e + R^e\left(\widehat{\mathbf{x}}^e\right) = \mathbf{f}^e, \tag{2.5}$$

where $\widehat{\mathbf{x}}^e = \begin{bmatrix} (\mathbf{x}_1^e)^T & (\mathbf{x}_2^e)^T & \ldots & \left(\mathbf{x}_{n-1}^e\right)^T & (\mathbf{x}_n^e)^T \end{bmatrix}^T$ and:

$$\mathbf{M}^e = \int_{\Omega_0^e} \mathbf{N}^{eT} \rho_0 \mathbf{N}^e dV, \quad R^e = \int_{\Omega_0^e} \mathbf{B}^{eT} S(\mathbf{N}^e \widehat{\mathbf{x}}^e) dV, \tag{2.6}$$

$$\mathbf{f}^e = \int_{\Omega_0^e} \mathbf{N}^{eT} \mathbf{b}_0 dV + \int_{\partial\Omega_0^e} \mathbf{N}^{eT} \bar{\mathbf{s}} dA, \tag{2.7}$$

with $I \in \mathbb{R}^{3\times3}$ and $\mathbf{N}^e = [\mathbf{N}_1^e I \quad \mathbf{N}_2^e I \cdots \mathbf{N}_n^e I] \in \mathbb{R}^{3\times3n}$, $\mathbf{B}^e = [\mathbf{B}_1^e \quad \mathbf{B}_2^e \cdots \mathbf{B}_n^e]$ with $\mathbf{B}_i^e \in \mathbb{R}^{9\times3}$ which first column is $\begin{bmatrix} \mathbf{N}_{i,1}^e & 0 & 0 & \mathbf{N}_{i,2}^e & 0 & 0 & \mathbf{N}_{i,3}^e & 0 \end{bmatrix}^T$ and the other ones are cyclic permutation of these elements ($\mathbf{N}^e_{i,j} \equiv \partial \mathbf{N}^e_i / \partial x_j$). Also, we have $\Omega^e_0 \subset \Omega_0$ is the regular region that is occupied by the element in the reference configuration, $\bar{\mathbf{s}}$ is the surface traction at the boundary of Ω^e_0, and $S(\mathbf{N}^e \widehat{\mathbf{x}}^e)$ represents the components of the divergence of the stress tensor σ which is symmetric with components: σ_{x_1,x_1}, σ_{x_2,x_2}, σ_{x_3,x_3}, σ_{x_1,x_2}, σ_{x_1,x_3}, σ_{x_2,x_3}.

After the element level equations are assembled, the resulting system has the general form:

$$\mathbf{M}_{FEM}\ddot{\widehat{\mathbf{x}}} + R\left(\widehat{\mathbf{x}}\right) = \mathbf{f}, \tag{2.8}$$

which is a system of ordinary differential equations. In its most general formulation the matrix $\mathbf{M}_{FEM}$ is positive definite, squared, and dense as it comes from the hessian of kinetic energy expressed using shape functions (ZIENKIEWICZ; TAYLOR; ZHU, 2005). The off-diagonal terms help to preserve the linear and angular momentum and give inertial contributes not depending on the considered node.

If the displacement gradient is small and the residual stress in reference configuration vanishes, then the relationship between the stress (σ) and the strain (ε) of the material can be approximated by linearization: $\sigma = G\varepsilon$, where G is a matrix parameterized by the Lame's constants, which are characteristics of the material (see (ZIENKIEWICZ; TAYLOR; ZHU, 2005; SADD, 2009) for details). In this case, we can show that equation (2.8) can be written as:

$$\mathbf{M}_{FEM}\ddot{\widehat{\mathbf{u}}} + \mathbf{K}_{FEM}\widehat{\mathbf{u}} = \mathbf{f}, \tag{2.9}$$

where $\widehat{\mathbf{u}}^e = \widehat{\mathbf{x}}^e - \widehat{\mathbf{x}}^e_0$; that means, the displacement respect to the reference configuration. The K_{FEM} is the stiffness matrix of the linearized FEM. Behind the expression (2.9) there is a simple definition of the elastic energy given by the quadratic function:

$$E_p\left(\widehat{\mathbf{u}}\right) = \frac{1}{2}\left(\widehat{\mathbf{u}}\right)^T \mathbf{K}_{FEM}\left(\widehat{\mathbf{u}}\right). \tag{2.10}$$

Also, we can introduce a dissipative term that is a linear function of $\dot{\widehat{u}}$, generating the model:

$$\mathbf{M}_{FEM}\ddot{\widehat{\mathbf{u}}} + \mathbf{D}_{FEM}\dot{\widehat{u}} + \mathbf{K}_{FEM}\widehat{\mathbf{u}} = \mathbf{f}, \tag{2.11}$$

where $\mathbf{D}_{FEM}$ is the finite element damping matrix, which is commonly chosen by using the generalized Rayleigh model (LLOYD et al., 2008):

$$\mathbf{D}_{FEM} = \mathbf{M}_{FEM}\sum_{i=0}^{L}\gamma_i\left(\mathbf{M}^{-1}_{FEM}\mathbf{K}_{FEM}\right)^i.$$

The solution of a general continuum problem by the finite element method always follows an orderly step-by-step process. In general, these step-by-step are organized into a sequence of activities which are: a) discretization of continuous body; b) selection of a proper interpolation or displacement model; c) derivation of element equations; d) assembly of element equations to obtain the overall equations; e) introduction of boundary conditions and solution for the unknown nodal displacements. The FEM literature (QUEK; LIU, 2003; RAO, 2004; REDDY, 2005; ZIENKIEWICZ; TAYLOR; ZHU, 2005; LOGAN,

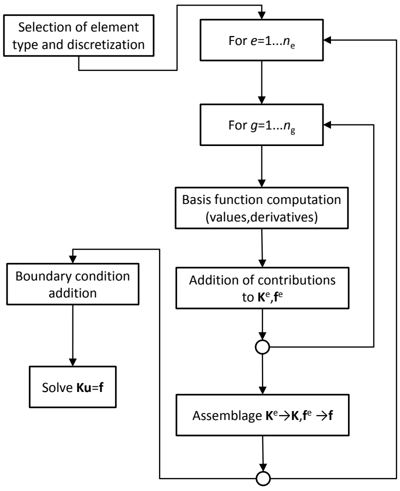

Selection of element type and discretization
For e=1...n_e
For g=1...n_g
Basis function computation (values,derivatives)
Addition of contributions to $\mathbf{K}^e,\mathbf{f}^e$
Assemblage $\mathbf{K}^e \rightarrow \mathbf{K}, \mathbf{f}^e \rightarrow \mathbf{f}$
Boundary condition addition
Solve $\mathbf{Ku}=\mathbf{f}$

Figure 2.3: Flowchart of activities for the solution of general continuum problem by FEM in equilibrium static scenario.

2011) presents detailed discussion about these steps. However, given their importance for this work, we have summarized them through the flowchart in Figure 2.3 which is an adaptation made from the reference (COTTRELL; HUGHES; BAZILEVS, 2009).

The flowchart in Figure 2.3 has as first process to gather some tasks which involves dividing the body into an equivalent system of finite elements with associated nodes and choosing the most appropriate element type to model most closely the actual physical behavior. The choice of element type used in a finite element analysis carries implicity the selection of displacement model. In other words, that mean the element type indirectly controls the selection of interpolation function which in turn define displacement model. In general, the displacement model is taken in the form of a polynomial and linear, quadratic, and cubic polynomials are frequently used as functions because they are simple to work with in finite element formulation and an easy and systematic method of generating interpolation functions of any order can be achieved by simple products of Lagrange polynomials which has been the preferred choice in classical FEM (RAO, 2004; ZIENKIEWICZ; TAYLOR; ZHU, 2005).

After this first processing step, the stiffness matrix and force vector of each element are calculated and assembled into the global structures (global stiffness matrix and force vector). We can see two loops to perform these tasks: the most internal loop which adds contributions calculated by each integration point (quadrature points) to element structures; and the most external loop which assembled local element structures into global ones. Concluded the assembly process the boundary conditions are introduced and finally the system is solved.

Once chosen the displacement model to linear or higher order polynomial (quadratic, cubic, etc), the discretization of the region (or domain) can be improved by two methods. In the first method, known as the *r-method*, the locations of the nodes are altered without changing the total number of elements. In the second method, known as the *h-method*, the number of elements is increased. On the other hand, if improvement in accuracy is sought by increasing the order of the interpolation of polynomial, the method is known as the *p-method* (RAO, 2004).

Problems involving curved boundaries cannot be modeled satisfactorily by using straight-sided elements (linear elements) (RAO, 2004). As demonstrated in (ZIENKIEWICZ; TAYLOR; ZHU, 2005), the using of higher order elements can significantly improve both the displacements and stresses obtained. Besides, fewer higher order elements are needed to achieve the same degree of accuracy in the final results. In this context, a family of elements known as "isoparametric" elements are widely used.

In isoparametric formulation, the term isoparametric is derived from the use of the same interpolation function (named shape function) N to define the element's geometric shape and to define the displacements within the element. Isoparametric element equations are formulated using a natural (or intrinsic) coordinate system (ξ, η, ζ), usually dimensionless with range $(\xi, \eta, \zeta) = (\pm 1, \pm 1, \pm 1)$, that is defined by element geometry and not by the element orientation in the global-coordinate system. There is a relationship (called a transformation mapping) between the natural coordinate system (ξ, η, ζ) and the global coordinate system (x, y, z) for each element of a specific structure, and this relationship must be used in the element equation formulations (RAO, 2004; LOGAN, 2011). Thus, the isoparametric (ξ, η, ζ) and global reference systems (x, y, z) are related by the following elementary equation:

$$\begin{aligned} x(\xi,\eta,\zeta) &= \sum_{i=1}^{n} N_i^e(\xi,\eta,\zeta)\, x_i^e, \\ y(\xi,\eta,\zeta) &= \sum_{i=1}^{n} N_i^e(\xi,\eta,\zeta)\, y_i^e, \\ z(\xi,\eta,\zeta) &= \sum_{i=1}^{n} N_i^e(\xi,\eta,\zeta)\, z_i^e, \end{aligned} \tag{2.12}$$

These equations are used to rewrite the integrals in equation (2.6) in terms of the natural system as

$$\mathbf{M^e} = \int_{-1}^{1} \mathbf{N}^{eT} \rho_\mathbf{0} \mathbf{N}^e \det(J)\, d\xi d\eta d\zeta, \tag{2.13}$$

$$\mathbf{K^e} = \int_{-1}^{1} \mathbf{B}^{eT} S(\mathbf{N}^e \widehat{\mathbf{x}}^e) \det(J)\, d\xi d\eta d\zeta, \tag{2.14}$$

where the shape functions $\mathbf{N}^e$ depends on natural coordinate(ξ, η, ζ), and $\det(J)$ is the Jacobian determinant of the mapping from natural coordinate system (ξ, η, ζ) to the global coordinate system (x, y, z).

2.3 BASIC B-SPLINE THEORY

The isogeometric approach has been developed using the NURBS which are built from B-Spline and so a discussion of B-Spline is a natural starting point for their investigation.

The sections 2.3-2.4 give a very brief overview about B-Spline and NURBS. The interested reader in NURBS theory may consult the introductory text in (PIEGL, 1991) or for a more detailed treatment can resort to the books(PIEGL; TILLER, 1997; FARIN, 1997).

A B-Spline entity is a generalization of the curve, surface or solid of Bezier defined by a linear combination of control points and basis functions over a parametric space wherein the basis functions are called B-Spline (short for basis spline) (FARIN, 1997; PIEGL; TILLER, 1997). The parametric space is divided into intervals and the B-Spline is defined piecewise on these intervals, with certain continuity requirements between the intervals. Thus, B-Spline functions are instances of piecewise polynomial functions associated with each interval. These interval are gathered under the set of coordinates in the parametric space known as knot vector. Due to its relevance in the B-spline theory, we can define it as

Definition 1. A knot vector is a finite, monotonically increasing sequence of real numbers express by

$$\Xi = (\xi_1, \ldots, \xi_m)\,,\ \xi_i \le \xi_{i+1},\ \forall i = \{1, \ldots, m-1\}\,. \tag{2.15}$$

Each item ξ_i, $i \in \{1, \ldots, m\}$ is referred as *knot.* A knot ξ_i is termed interior knot if $(\xi_1 < \xi_i)$ and $(\xi_i < \xi_m)$. The knot ξ_i has multiplicity $l = |\{\xi_j \mid j \in \{1, \ldots, m\} \wedge \xi_i = \xi_j\}|$ such that l is the occurrences number of knot ξ_i in Ξ. A *knot span* is the interval between two consecutive knots (ξ_i, ξ_{i+1}) for $\forall i = \{1, \ldots, m-1\}$ which is termed empty knot span when $\xi_i = \xi_{i+1}$.

Having defined the knot vector, we can proceed with the B-Spline basis functions definition.

Definition 2. Let $p \ge 0$ and $\Xi = (\xi_1, \ldots, \xi_m)$ be the polynomial order [1] and knot vector from the B-Spline basis functions, respectively. By considering the multiplicity of any interior knot is at most p and number of basis functions is $n = m - p - 1$, the n univariate B-Spline basis functions $N_{i,p}(\xi) : [\xi_1, \xi_m] \to \mathbb{R}$, $i = 1, \ldots, n$ are defined by the Cox-de Boor recursion formula (BOOR, 1978) as follows:

$$N_{i,0}(\xi) = \begin{cases} 1, & if\, \xi_i \le \xi < \xi_{i+1} \\ 0, & otherwise, \end{cases} \tag{2.16}$$

when $p = 0$ and for a polynomial order $p \ge 1$

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi). \tag{2.17}$$

The knot vector has a significant influence in the B-Spline generated. In general, one can use three types of knot vectors: uniform, open uniform (or just open) and nonuniform.

[1] The convention we will adopt is that the order $p = 0, 1, 2, 3$, etc., refers to constant, linear, quadratic, cubic, etc., piecewise polynomials, respectively. What we refer to as "order" is usually referred to as "degree" in the computational geometry literature.

Definition 3. The knot vector $\boldsymbol{\Xi}$ is said to be open if the multiplicity of a knot is at most p, except the end knots which have multiplicity $(p+1)$. Uniform knot vectors satisfies $\xi_{i+1} - \xi_i = \Delta\xi = const.$, for $i = 1, ..., m-1$. Uniform knot vectors yield periodic uniform basis functions. An open uniform knot vector has also the property $\xi_{i+1} - \xi_i = \Delta\xi$ for internal knots but it has multiplicity of knot values at the ends equal to $(p+1)$. Finally, nonuniform knot vectors may have either unequally spaced ($\xi_{i+1} - \xi_i = \Delta\xi$) and/or multiple knot values at the ends or even for the internal knots.

From (2.16) and (2.17), one can verify that B-Spline basis functions possess the following properties (PIEGL; TILLER, 1997):

a) Partition of unity: $\sum_{i=1}^{n} N_{i,p}(\xi) = 1$.

b) Pointwise nonnegativity: $N_{i,p}(\xi) \geq 0 \ \forall \xi$.

c) Linear independence: $\sum_{i=1}^{n} \alpha_k N_{i,p}(\xi) = 0 \iff \alpha_k = 0, k = 1, \ldots, n$.

d) Local support: $N_{i,p}(\xi) = 0$ if $\xi \notin [\xi_i, \xi_{i+p+1}]$.

2.3.1 Multivariate B-Spline Basis Functions

Multivariate B-Spline basis functions are defined by tensor products of univariate B-Spline basis functions. In order to build multivariate B-Spline basis functions, we introduce the d-dimensional multi-index $\mathbf{i} \in \mathbb{N}^d$.

Definition 4. Let $\boldsymbol{\Xi} = \left(\Xi^{(1)}, \ldots, \Xi^{(d)}\right)$ be the knot vectors, one for each dimension such that

$$\boldsymbol{\Xi} = \left(\Xi^{(1)} = \left(\xi_1^{(1)}, \ldots, \xi_{m_1}^{(1)}\right), \ldots, \Xi^{(d)} = \left(\xi_1^{(d)}, \ldots, \xi_{m_d}^{(d)}\right)\right),$$

from which it is defined the parametric space as

$$\hat{\Omega} = \left(\xi_1^{(1)}, \xi_{m_1}^{(1)}\right) \times \ldots \times \left(\xi_1^{(d)}, \xi_{m_d}^{(d)}\right). \tag{2.18}$$

Definition 5. Let $\mathcal{U} = (u_1, \ldots, u_d)$ be the upperbound multi-index for each dimension and let $\mathcal{I}(\mathcal{U})$ be the function that generates a set of tuples for d-dimensional multi-index $\mathbf{i} \in \mathbb{N}^d$ defined by

$$\mathcal{I}(\mathcal{U}) = \{(i_1, \ldots, i_d) \in \{(1 \ldots u_1) \times \ldots \times (1 \ldots u_d)\}, u_s \in \mathcal{U}, s = 1 \ldots d\}.$$

Definition 6. Let upperbound multi-index $\mathcal{U}_b = (u_1 = n_1, \ldots, u_d = n_d)$ be the number of basis functions along each dimension. Thus, we define index space for the basis functions as

$$\mathcal{I}_b = \mathcal{I}(\mathcal{U}_b),$$

where the subscript b means "basis".

Definition 7. Let upperbound multi-index $\mathcal{U}_k = (u_1 = m_1, \ldots, u_d = m_d)$ be the number of knots for each knot vector along each dimension. Thus, we define index space for the knot vectors as

$$\mathcal{I}_k = \mathcal{I}(\mathcal{U}_k),$$

where the subscript k means "knot", and $m_1 = n_1 + p_1 + 1, \ldots, m_d = n_d + p_d + 1$.

Definition 8. We define corresponding d-variate B-Spline basis functions as

$$N_{\mathbf{i},\mathbf{p}}\left(\boldsymbol{\xi}\right) = \prod_{s=1}^{d} N_{i_s,p_s}\left(\xi^{(s)}\right) = N_{i_1,p_1}^{(1)}\left(\xi^{(1)}\right) \cdot \ldots \cdot N_{i_d,p_d}^{(d)}\left(\xi^{(d)}\right), \tag{2.19}$$

where $\mathbf{i} \in \mathcal{I}_b$ is d-dimensional multi-index, $\boldsymbol{\xi} = \left(\xi^{(1)}, \ldots, \xi^{(d)}\right)$ are the coordinates value along each parametric direction, and $\mathbf{p} = (p_1, \ldots, p_d)$ denote the polynomial order along each parametric direction.

2.3.2 B-Spline Geometries

Once defined the B-Spline basis functions, we can consider the creating of several geometry using the functions basis discussed above.

Definition 9. Let $N_{\mathbf{i},\mathbf{p}}\left(\boldsymbol{\xi}\right)$ be a family of d-variate B-Spline basis functions where $\mathbf{i} \in \mathcal{I}_b$, $\boldsymbol{\xi} = \left(\xi^{(1)}, \ldots, \xi^{(d)}\right)$ and $\mathbf{p} = (p_1, \ldots, p_d)$ are the coordinate values and the polynomial orders along each parametric direction. Given a control mesh $\mathbf{p_i} \in \mathbb{R}^3$, the generalized geometry mapping $\mathbf{g} : \hat{\Omega} \to \Omega \subset \mathbb{R}^d$ is defined by

$$\mathbf{g}\left(\boldsymbol{\xi}\right) = \sum_{\mathbf{i} \in \mathcal{I}_b} N_{\mathbf{i},\mathbf{p}}\left(\boldsymbol{\xi}\right) \mathbf{p_i}, \tag{2.20}$$

where $\hat{\Omega}$ and $\Omega \subset \mathbb{R}^d$ represent the parametric and physical spaces, respectively.

For $d = 1$, the geometry mapping $\mathbf{g}$ defines the B-Spline curve. Thus, we have the B-Spline parametrization given by

$$\mathcal{I}_b = \left\{i_1 \in \{(1 \ldots n_1)\}\right\}, \ \mathbf{p} = p_1, \ \boldsymbol{\Xi} = \Xi^{(1)}, \ \boldsymbol{\xi} = \xi^{(1)}.$$

The B-Spline surface is defined by $\mathbf{g}$ when $d = 2$ and the B-Spline parametrization is given by

$$\begin{aligned}
\mathcal{I}_b = \left\{(i_1, i_2) \in \{(1 \ldots n_1) \times (1 \ldots n_2)\}\right\}, \\
\mathbf{p} = (p_1, p_2), \\
\boldsymbol{\Xi} = \left(\Xi^{(1)}, \Xi^{(2)}\right), \\
\boldsymbol{\xi} = \left(\xi^{(1)}, \xi^{(2)}\right).
\end{aligned}$$

The geometry mapping $\mathbf{g}$ define a B-Spline solid for $d = 3$, where the B-Spline parametrization is given by

$$\begin{aligned}\mathcal{I}_b = \{(i_1, i_2) \in \{(1 \ldots n_1) \times (1 \ldots n_2) \times (1 \ldots n_3)\}\}, \\ \mathbf{p} = (p_1, p_2, p_3), \\ \boldsymbol{\Xi} = \left(\Xi^{(1)}, \Xi^{(2)}, \Xi^{(3)}\right), \\ \boldsymbol{\xi} = \left(\xi^{(1)}, \xi^{(2)}, \xi^{(3)}\right).\end{aligned}$$

2.4 NON-UNIFORM RATIONAL B-SPLINE

It is not always possible to model exactly all geometric entities by piecewise polynomials such as B-Spline. Many important geometric entities can only be obtained through a projective transformation of a B-Spline entity in $\mathbb{R}^{d+1}$ yielding a rational B-spline into $\mathbb{R}^d$. Examples of geometric entities are conic sections, such as circles and ellipses, can be exactly constructed by projective transformations of B-Spline quadratic curves.

2.4.1 Rational B-Spline Curves

Definition 10. Let $\widetilde{\mathbf{p}}_i$ be a control points in the four-dimensional homogeneous coordinate space such that

$$\widetilde{\mathbf{p}}_i = \begin{pmatrix} w_i \mathbf{p}_i \\ w_i \end{pmatrix}, \quad i = 1 \ldots n,$$

we can define a polynomial B-Spline curve in the four-dimensional homogeneous as

$$\widetilde{\mathbf{c}}(\xi) = \sum_{i=1}^{n} \begin{pmatrix} w_i \mathbf{p}_i \\ w_i \end{pmatrix} N_{i,p}(\xi). \tag{2.21}$$

By projection in the three-dimensional space we obtain the rational curve:

$$\mathbf{c}(\xi) = \frac{\sum_{i=}^{n} w_i N_{i,p}(\xi) \mathbf{p}_i}{\sum_{j=1}^{n} w_j N_{j,p}(\xi)} = \sum_{i=1}^{n} R_{i,p}(\xi) \mathbf{p}_i, \tag{2.22}$$

where $R_{i,p}(\xi)$ are the rational B-spline functions given by:

$$R_{i,p}(\xi) = \frac{w_i N_{i,p}(\xi)}{\sum_{i=1}^{n} w_i N_{i,p}(\xi)}. \tag{2.23}$$

This function defines a curve of class C^{p-1} in $\mathbb{R}^3$, which is called a B-Spline curve. The points $\mathbf{p}_i$ are called control points and the corresponding polygon is the defining polygon or control polygon. Important properties about these curves are:

1. End points interpolation: in the case of open knot vector we have $\mathbf{c}(\xi_1) = \mathbf{p}_1$ and $\mathbf{c}(\xi_{n+p+1}) = \mathbf{p}_n$.

2. Affine Invariance: If $\psi(r) = Ar + v$ is an affine transformation then $\psi(c(\xi)) = \sum_{i=1}^{n} N_{i,p}(\xi)\,\psi(\mathbf{p}_i)$.

3. Strong convex hull property: the curve belongs to the convex hull of its control polygon.

Remark. When the rational B-spline functions are computed with non-periodic and non-uniform knot vector they are named Non-Uniform Rational B-Splines (NURBS).

2.4.2 B-Spline Rational Surfaces and Solids

Analogous to the B-Spline basis functions the multivariate NURBS basis functions also can be defined with multi-index. Thereunto, we begin by defining the generalized multivariate NURBS basis functions.

Definition 11. Let $\boldsymbol{\Xi} = \left(\Xi^{(1)}, \ldots, \Xi^{(d)}\right)$ be the knot vectors, one for each dimension as defined in (4). We define corresponding d-variate NURBS basis functions as

$$R_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi}) = \prod_{s=1}^{d} N_{i_s,p_s}^{(s)}\left(\xi^{(s)}\right) = \frac{w_{\mathbf{i}} N_{i_1,p_1}^{(1)}\left(\xi^{(1)}\right) \cdot \ldots \cdot N_{i_d,p_d}^{(d)}\left(\xi^{(d)}\right)}{\sum_{\mathbf{i}'} w_{\mathbf{i}'} N_{i'_1,p_1}^{(1)}\left(\xi^{(1)}\right) \cdot \ldots \cdot N_{i'_d,p_d}^{(d)}\left(\xi^{(d)}\right)}, \tag{2.24}$$

where $\mathbf{i}, \mathbf{i}' \in \mathcal{I}_b$ are d-dimensional multi-indexes, $\boldsymbol{\xi} = \left(\xi^{(1)}, \ldots, \xi^{(d)}\right)$ are the coordinates value along each parametric direction, and $\mathbf{p} = (p_1, \ldots, p_d)$ denote the polynomial order along each parametric direction.

Definition 12. The NURBS curve, surface and solid are defined analogously to the Definition 9. To generate NURBS geometry just replace the basis functions in the expression (2.20) by NURBS basis functions , which leads us to

$$\mathbf{g}(\boldsymbol{\xi}) = \sum_{\mathbf{i} \in \mathcal{I}_b} R_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi})\,\mathbf{p}_{\mathbf{i}}. \tag{2.25}$$

Remark. In almost all practical circumstances, the modeling of complex topologies with conventional NURBS need several tensor product pieces which should be patched together, whence comes the idea of NURBS patches. In other words, a NURBS patch is a particular set of knot vectors, polynomial order, and control points associated with a specific region of the model. Each patch has its own parameter space. Large geometries are frequently built from many patches. When two patches meet, the control points coming from each side must be identical along the interface where they meet, and the corresponding knot vectors must be identical as well. Under these conditions, only C^0-continuity of the basis is achieved across the patch boundaries (BAZILEVS et al., 2010).

Assumption 13. *Hereinafter, we assume for the remainder of this thesis that all used NURBS geometries will be built from a single NURBS patch.*

2.5 ISOGEOMETRIC ANALYSIS

To perform the behavior simulation of deformable objects we can think about an engine which is composed by three linked parts: the geometric model, dynamic model and rendering module. The former can be realized in the context of parametric frameworks like Non-uniform Rational B-Spline (NURBS) (PIEGL; TILLER, 1997; FARIN, 1997). The dynamic model needs physic models that incorporate dynamic quantities like velocity, mass and force distributions, into an evolution equation that governs the shape deformation (ERLEBEN et al., 2005). The latter includes global/local illumination techniques to generate the scene with the desired realism (AKENINE-MöLLER; HAINES; HOFFMAN, 2008). Of course, from what has been presented so far that this work is focused only on the first two components.

The use of NURBS as geometric model in the simulation context of deformable objects can bring improvements due to the features that NURBS offers. The NURBS is a mathematical framework commonly used for generating and representing curves, surfaces and solids (PIEGL; TILLER, 1997). It offers an unified mathematical basis to describe analytic and free-form shapes with great flexibility and precision. It became a standard for CAD (Computer Aided Design) systems due to its excellent mathematical, numeric and algorithmic properties.

In turn, try to gather NURBS with a dynamic model in the context of deformable model it is not something recent. The researchers have explored ways to couple the NURBS geometric representation with physic modeling. One of the seminal works is the Dynamic NURBS (D-NURBS) method developed by Terzopoulos and Qin (TERZOPOULOS; QIN, 1994) that aimed at applying the deformable modeling approach to shape design.. In D-NURBS, the NURBS control points and the weights play the role of generalized coordinates, and the dynamical equations are derived from the Hamiltonian principle. The deformation of a NURBS body is described by displacement the control points and weights.

However, it was the need to unify the analysis process of the FEM with higher accuracy of geometry representation that has linked the geometric model of NURBS with finite element analysis. For this, the isogeometric analysis (IGA) has been introduced in (HUGHES; COTTRELL; BAZILEVS, 2005) as an alternative to FEM-based method for the analysis of problems governed by partial differential equations, inspired by the desire to unify the fields of computer aided geometric design (CAGD) and FEM.

The main feature of IGA is the ability to maintain the same exact description of the computational domain geometry throughout the analysis process, including refinement. In the words of Bazilevs (BAZILEVS et al., 2013), the IGA is an inherently higher-order accurate technique which uses basis functions emanating from CAGD, such as B-splines, NURBS, T-splines, subdivision surfaces, etc., instead of traditional C^0-continuous Lagrange finite element interpolatory polynomials (TEMIZER; WRIGGERS; HUGHES, 2011). These basis functions are of higher-order continuity than in the standard FEM. This additional continuity is a distinguishing feature of IGA and it is beneficial in many applications. Due to the increased order and smoothness of the underlying basis functions, IGA is able to tolerate larger levels of mesh distortion than standard finite elements with-

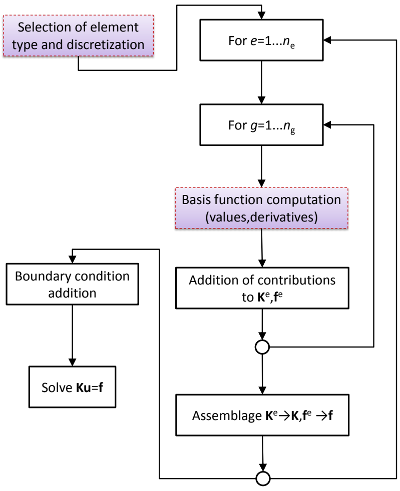

Selection of element type and discretization
For e=1...n_e
For g=1...n_g
Basis function computation (values,derivatives)
Boundary condition addition
Addition of contributions to K^e,f^e
Solve Ku=f
Assemblage K^e→K,f^e →f

Figure 2.4: Flowchart of activities for the solution of general continuum problem by IGA in equilibrium static scenario

out compromising solution accuracy. These are features that we can explore to improve the quality of MSM derivation process and the accuracy of geometry representation.

In the words of Cottrell *et al* (COTTRELL; HUGHES; BAZILEVS, 2009), the isogeometric analysis has a structure quite similar to classical FEM. Such similarity can be observed by the orderly step-by-step that the finite element method follows to solve a general continuum problem. In section 2.2, we summarized these general steps through the flowchart in Figure 2.3. The mandatory modification to the flowchart such that it could be used into isogeometric analysis is shown in Figure 2.4. As can be seen, the flowchart structure is identical. However, we have highlighted in gray and red dotted lines those activities that must be modified to comply isogeometric analysis requirements. The activities related to selection of elements, discretization and basis functions computation due to isoparametric concept depends on the choice of basis functions. Therefore, in short, the major difference between classical FEM and IGA is the basis functions being used.

Since exists an strong connection between classical FEM and IGA it is expected that the conceptual itens of the IGA also keep some relation with the FEM. In FEM there is one notion of a mesh and one notion of an element as seen in section 2.2. The key idea in FEM is to subdivide the object domain into subdomains or finite elements with each subdomain represented by a set of element equations which are systematically recombined into a global system to compute the fields variable in question (e.g., displacement, velocity, temperature, etc.). All these items are also present in NURBS-based IGA model, however they are defined and treated differently.

Lagrange interpolation polynomials have been the standard choice for basis functions in finite element method from almost the very beginning of finite element research. They are the overwhelming choice whether the preferred metric is their prevalence in commercial codes or in research articles (BENSON et al., 2010). On the other hand, basically all functions used in CAD could be used as basis for isogeometric analysis, provided that they fulfill the necessary conditions for basis functions in modeling tasks, such as local support and partition of unity (COTTRELL; HUGHES; BAZILEVS, 2009; NGUYEN; BORDAS, 2015). Non-uniform Rational B-Splines are the most widespread technology in today's CAD programs and they fulfill the necessary conditions mentioned above, therefore they are adopted for analysis. In the next sections, we present more detail of the theoretical apparatus of the isogeometric analysis which is indispensable to a better understanding of NURBS-based IGA framework.

2.5.1 Element Formulation

Undoubtedly, the core of the finite element method is the idea of element. An element represents the elemental unit of solution in the solving of global problem. It holds a number of attributes that can influence the solution quality. Therefore, the study of element properties are extremely important for proper application of the finite element method. Given such importance, it is compulsory to understand how the element is defined in the context of isogeometric analysis.

Likewin in the finite element method, a NURBS-based IGA element is formed by a set of nodes and corresponding interpolation functions. The nodes from FEM have their counterpart in the NURBS control points as well as interpolation functions have matching with NURBS basis functions. Hence, the control points constitutes the degrees of freedom in each element such that the analysis and boundary conditions are applied to them. However, a NURBS-based IGA element has characteristic and formulation differents from those found in FEM elements. To understand these differences we will develop a formulation for the NURBS-based IGA.

Definition 14. Let $\Xi = \left(\Xi^{(1)}, \dots, \Xi^{(d)}\right)$ be the knot vectors, one for each dimension as set out in Definition 4 and let $\mathcal{I}_k$ be the index space corresponding to Ξ. We denote a tensor product mesh which divides the parametric domain $\hat{\Omega}$ as a set built from the following formation rule

$$\hat{\mathcal{K}} = \left\{ (\Xi_{\mathbf{i}}, \Xi_{\mathbf{i}+1}) = \left(\xi^{(1)}_{i_1}, \xi^{(1)}_{i_1+1}\right) \times, \dots, \times \left(\xi^{(d)}_{i_d}, \xi^{(d)}_{i_d+1}\right) \mid \mathbf{i} \in \mathcal{I}_k \wedge \mathbf{i} \in \{\mathbf{i} - 1\} \right\}, \tag{2.26}$$

where each d-dimensional $\kappa_{\mathbf{i}} \in \hat{\mathcal{K}}$ is termed parametric element. These elements are typically a line segment in 1-dimensional problems, a quadrangle in 2-dimensional and a hexahedron in 3-dimensional case.

Notice that the ordered pair forming the Cartesian product in expression (2.26) are knot spans in their respective knot vectors (see Definition 1). Therefore, we must be aware to ocurrences of empty knots span which could cause line, quadrangle or hexahedron to have dimensional size (length, area, volume) equal to zero. To remove parametric elements with these characteristics, we define a subset from $\hat{\mathcal{K}}$ as

$$\tilde{\mathcal{K}} = \left\{ \kappa_{\mathbf{i}} \in \hat{\mathcal{K}} \mid \xi^{(s)}_{i_s} = \xi^{(s)}_{i_s+1}, s = 1 \dots d \right\},$$

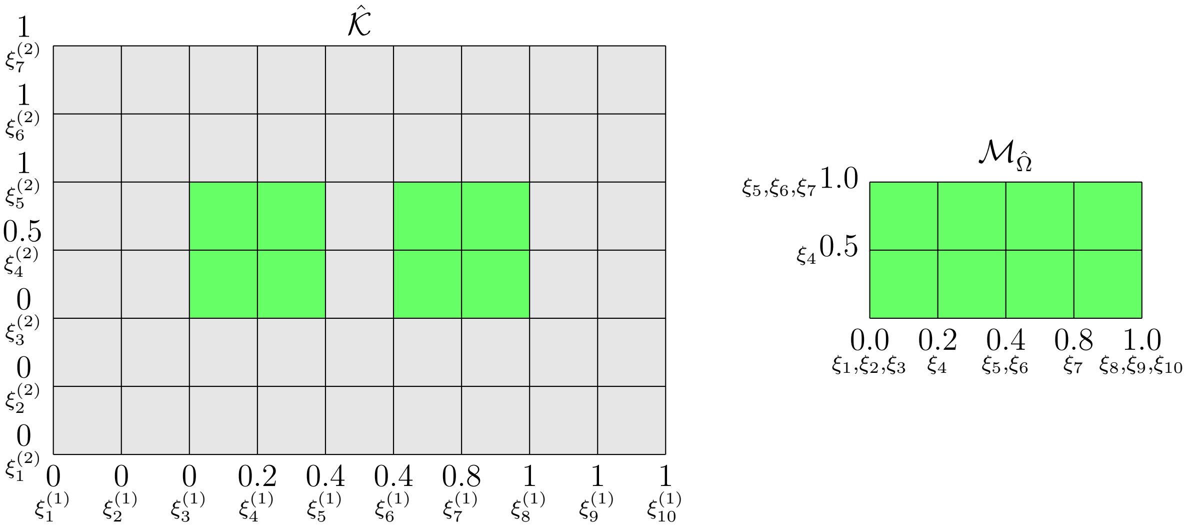


Figure 2.5: The parametric mesh $\hat{\mathcal{K}}$ dividing the parametric domain $\hat{\Omega}$.

which allow us to define the set $\mathcal{M}_{\hat{\Omega}}$ as

$$\mathcal{M}_{\hat{\Omega}} = \hat{\mathcal{K}} - \tilde{\mathcal{K}}. \tag{2.27}$$

Remark. The set $\mathcal{M}_{\hat{\Omega}}$ is called parametric mesh on $\hat{\Omega}$ containing only elements with proper dimensional size. We termed $\kappa_{\mathbf{i}} \in \tilde{\mathcal{K}}$ as empty parametric element due to the fact that it has empty knot span. Otherwise when $\kappa_{\mathbf{i}} \in \mathcal{M}_{\hat{\Omega}}$, it is termed non-empty parametric element.

Example. To further clarify these definitions, let us consider the 2-dimensional case with the following NURBS parameters:

$$\Xi^{(1)} = \left\{ \underset{\xi_1^{(1)}}{0}, \underset{\xi_2^{(1)}}{0}, \underset{\xi_3^{(1)}}{0}, \underset{\xi_4^{(1)}}{0.2}, \underset{\xi_5^{(1)}}{0.4}, \underset{\xi_6^{(1)}}{0.4}, \underset{\xi_7^{(1)}}{0.8}, \underset{\xi_8^{(1)}}{1}, \underset{\xi_9^{(1)}}{1}, \underset{\xi_{10}^{(1)}}{1} \right\},$$

$$\Xi^{(2)} = \left\{ \underset{\xi_1^{(2)}}{0}, \underset{\xi_2^{(2)}}{0}, \underset{\xi_3^{(2)}}{0}, \underset{\xi_4^{(2)}}{0.5}, \underset{\xi_5^{(2)}}{1}, \underset{\xi_6^{(2)}}{1}, \underset{\xi_7^{(2)}}{1} \right\},$$

$$\mathcal{I}_k = \{(i_1, i_2) \in \{(1 \ldots m_1 = 10) \times (1 \ldots m_2 = 7)\}\}.$$

In Figure 2.5 is shown the parametric mesh $\hat{\mathcal{K}}$ that contained all elements $\kappa_{\mathbf{i}}$ including those with empty knot span. One can view knot labels and their respective values ξ_i which are distributed on parametric domain $\hat{\Omega}$. We pictured each $\kappa_{\mathbf{i}} \in \tilde{\mathcal{K}}$ (empty element) with gray color, and each $\kappa_{\mathbf{i}} \in \mathcal{M}_{\hat{\Omega}}$ (non-empty element) is pictured with green color.

In the region on the right of Figure 2.5 , we highlight the set $\mathcal{M}_{\hat{\Omega}}$. We gather the knot labels with multiplicity $l > 1$ under its reference value. Analyzing the distribution of knots values in $\mathcal{M}_{\hat{\Omega}}$ which is an immediate consequence of the formation rules for the sets $\hat{\mathcal{K}}$ and $\tilde{\mathcal{K}}$, we achieve to conclusion that lead us to a natural definition of NURBS-based IGA element.

Definition 15. Given the knot vectors $\boldsymbol{\Xi} = \left(\Xi^{(1)}, \ldots, \Xi^{(d)}\right)$ and the parametric space confined to $\hat{\Omega} = (0,1)^d$, the NURBS-based IGA element is defined by the cartesian product between the ordered pair of knots in each direction belonging to a non-empty knot span, namely:

$$\hat{\Omega}_e = \kappa_{\mathbf{i}} \in \mathcal{M}_{\hat{\Omega}}$$

where $\mathcal{M}_{\hat{\Omega}}$ the parametric mesh on $\hat{\Omega}$ containing only non-empty elements, and $\mathbf{i} \in \mathcal{I}_k$ is multi-index to index space of the knot vectors.

2.5.2 Discretization and Analysis

As seen in the Definitions (9) and (12), the B-spline and NURBS mappings transform coordinates in parameter space $\hat{\Omega}$ to physical space Ω. Applying the key idea in finite element method that is to subdivide the object domain into subdomains or finite elements, we can define these two spaces in terms of elements as

$$\hat{\Omega} = \bigcup_{e=1}^{n_e} \hat{\Omega}_e, \ \Omega = \bigcup_{e=1}^{n_e} \Omega_e, \tag{2.28}$$

where n_e is the total number of elements in the mesh $\mathcal{M}_{\hat{\Omega}}$, and Ω_e is an element in the physical space. Analogous to the expression (2.25), the geometric mapping $\mathbf{g} : \hat{\Omega}_e \to \Omega_e \subset \mathbb{R}^d$ can be obtained as

$$\mathbf{g}\left(\hat{\Omega}_e\right) = \sum_{\mathbf{i} \in \mathcal{I}_b} R_{\mathbf{i},\mathbf{p}}\left(\hat{\Omega}_e\right) \mathbf{p}_{\mathbf{i}}. \tag{2.29}$$

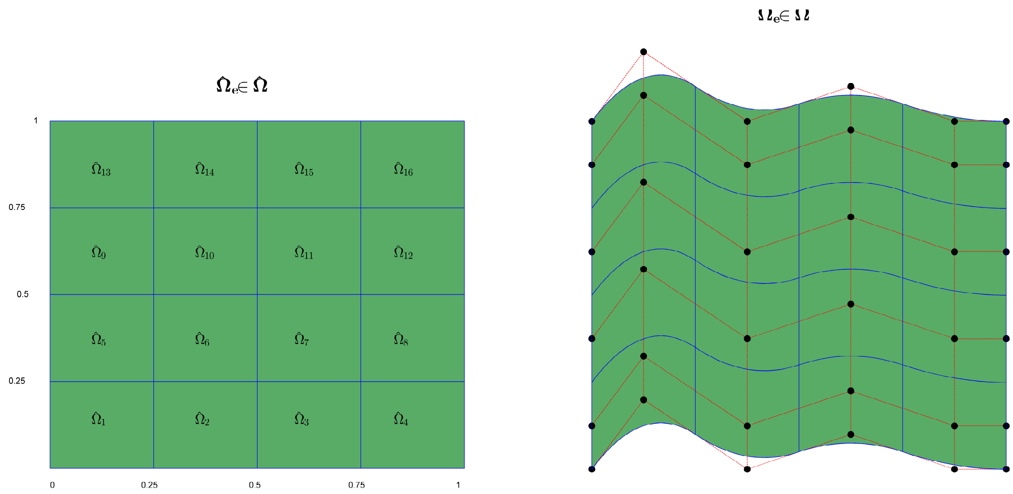
In a sense, the expression (2.28) reflects the discretization of parametric and physical spaces by the respective elements. In classical FEM, the discretization process generates a only mesh under which will take place the analysis process. In case of the expression (2.28) it is perceived two meshes: parametric and physical. Furthermore, by expression (2.29) we can conclude that each element $\hat{\Omega}_e$ has its image Ω_e in the physical space, ie, parametric and physical meshes have a dependent relationship in which we can add yet the spatial configuration of a third mesh: control mesh.

In Figure 2.6(a) is shown parametric mesh for knot vectors

$$\Xi^{(1)} = \Xi^{(2)} = \{0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1\}.$$

In Figure 2.6(b) it is illustrated the NURBS geometric mapping for the parametric space of Figure 2.6(a). Here, the control mesh is shown in red dotted lines with control points denoted by black circles. The non-interpolatory nature of control points can be seen through the delineation of the physical mesh (colored in green with blue lines indicating the elements boundary).

In the context of deformable models, the analysis process in FEM is related to obtain numerical solution of partial differential equations describing the continuum behavior of deformable objects. The main focus of the analysis, as mentioned in section 2.2, is to



(a) (b)

Figure 2.6: The geometry mapping $\mathbf{g}$ between parameter space (a) $\hat{\Omega}$ and physical space (b) Ω.

assemble the governing algebraic equations in matrix form and to compute the unknown values of the primary field variable (e.g., displacement, velocity, temperature, etc.). The assembly of the governing algebraic equations (see section 2.2) can be expressed in terms of summation as

$$\sum_{e=1}^{n_e} \left(\mathbf{M}^e \ddot{\widehat{\mathbf{u}}} + \mathbf{K}^e \widehat{\mathbf{u}} = \mathbf{f}^e \right),$$

where the summation symbol indicates the assembling procedure to evaluate the global system of equations, considering contributions at element level; $\mathbf{M}^e$, $\mathbf{K}^e$ are the mass and stiffness element matrices, respectively; and $\mathbf{f}^e$ is the force vector at element level. In static analysis, it is dropped the term with the mass matrix (as detailed in the flowchart in Figure 2.4) and equation becomes

$$\sum_{e=1}^{n_e} \left(\mathbf{K}^e \widehat{\mathbf{u}} = \mathbf{f}^e \right).$$

The Gaussian quadrature rule is applied to the evaluation of the element matrices during numerical integration. Integrals defining the matrix terms, which are initially defined in the physical space, are transferred to the parametric space and then to the space containing the quadrature points (parent space), where the numerical integration is actually performed.

The transformation from the physical space to the parent space is ilustraded in Figure 2.7, where Gaussian quadrature is carried out, is achieved by using a composition of two consecutive transformations: the physical space ($\mathbf{x}$) is transferred first to the parametric space ($\hat{\boldsymbol{\xi}}$) through a geometrical mapping and then to the parent space ($\tilde{\boldsymbol{\xi}}$) through

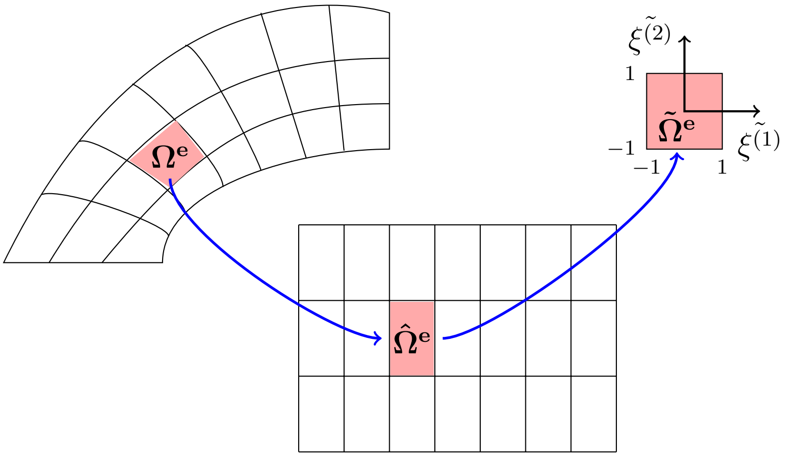


Figure 2.7: The spaces (physical, parametric and parent) utilized in the numerical integration of the element matrices

a second mapping, which is affine. Spatial derivatives of basis functions with respect to Cartesian coordinates are substituted by the corresponding derivatives with respect to the parametric coordinates as follows (ESPATH; BRAUN; AWRUCH, 2011):

$$\frac{\partial R}{\partial x_i} = \frac{\partial R}{\partial \widehat{\xi}_j} \frac{\partial \widehat{\xi}_j}{\partial x_i},$$

where the second term on the right-hand side of expression represents the inverse of the Jacobian matrix. Since the numerical integration is performed in the parent domain, the Jacobian determinant is evaluated with:

$$J = \left| \frac{\partial x_i}{\partial \tilde{\xi}_j} \right| = \left| \frac{\partial x_i}{\partial \widehat{\xi}_k} \frac{\partial \widehat{\xi}_k}{\partial \tilde{\xi}_j} \right|.$$

The parametric coordinates $\hat{\boldsymbol{\xi}} = \left(\hat{\xi}_1, \hat{\xi}_2, \hat{\xi}_3\right)$ related to the quadrature points defined in the parent space $\tilde{\boldsymbol{\xi}} = \left(\tilde{\xi}_1, \tilde{\xi}_2, \tilde{\xi}_3\right)$ can be obtained considering the NURBS multi-index $\mathbf{i} \in \mathcal{I}_b$ associated to element $\kappa_{\mathbf{i}} \in \mathcal{M}_{\hat{\Omega}}$, that is:

$$\hat{\boldsymbol{\xi}} = \hat{\boldsymbol{\xi}}_{\mathbf{i}} + \left(\tilde{\boldsymbol{\xi}} + 1\right) \frac{\hat{\boldsymbol{\xi}}_{\mathbf{i}+1} - \hat{\boldsymbol{\xi}}_{\mathbf{i}}}{2}.$$

To construct the IGA solution field over a NURBS patch, the isoparametric concept is applied by representing the discrete solution $\widehat{\mathbf{u}}$ in terms of the NURBS basis functions, which, in turn, are determined by the given geometry mapping

$$\widehat{\mathbf{u}}\left(\boldsymbol{\xi}\right) = \sum_{\mathbf{i}\in\mathcal{I}_b} R_{\mathbf{i},\mathbf{p}}\left(\boldsymbol{\xi}\right)\mathbf{u}_{\mathbf{i}},$$

where $\mathbf{u}_i$ are the control variables or degrees-of-freedom. Assuming the geometric mapping $\mathbf{g}$ (see Definitions 9 and 12) is invertible, these NURBS basis functions can be brought into the physical space by

$$R_{\mathbf{i},\mathbf{p}}\left(\mathbf{x}\right) = R_{\mathbf{i},\mathbf{p}}\left(\boldsymbol{\xi}\right)\circ\mathbf{g}^{-1}.$$

Thus, given the solution in the parametric space $\widehat{\mathbf{u}}\left(\boldsymbol{\xi}\right)$, its physical space counterpart may be defined as

$$\mathbf{u}\left(\mathbf{x}\right) = \sum_{\mathbf{i}\in\mathcal{I}_b} R_{\mathbf{i},\mathbf{p}}\left(\boldsymbol{x}\right)\mathbf{u}_{\mathbf{i}}.$$

2.6 CONCLUSIONS

Interactivity is an important requirement for applications such as animation, virtual environments, surgical training and games. These applications need to perform the simulation of mechanical behaviour in real-time; that means that one second of simulation corresponds to one second of real life.

The techniques for simulating deformable objects can be grouped into continuous and discrete models. The continuous methods are traditionally accurate relative to discrete methods due to its connection with continuum mechanics as can be seen in sections 2.1-2.2.

In this chapter we mainly describe two continuous models: the finite element method (FEM) and isogeometric analysis (IGA). These models can be used to perform mechanical simulation of deformable objects. In general, they cannot be applied to interactive simulations due to their high computational cost.

The described characteristics of two methods can be used to design discrete models so that continuous and discrete models behave similarly. The main of these characteristics is accuracy to simulate mechanical behaviour. However mesh refinement techniques and the use of higher order elements (isoparametric concept) also can be applied to improve both accuracy of geometry and solution of field variable.

The next chapter is about the discrete methods. In particularly, we describe the mass spring model (MSM) and how continuous models has been applied to design the parameters of discrete models.

Chapter 3

The MSM has been a discrete approach widely employed to simulation of deformable object which is simple to implement and can be faster then the continuous ones, and so, more suitable for real time and interactive application. However, obtaining parameters like stiffness coefficients still remains an open challenge in MSM design. In this chapter we present a simplified formulation to MSM together with the description to the parameterization strategies of the MSM.

DISCRETE MODELS

We can characterize a discrete deformable model as the one that does not use the physical constants directly related to the simulated material (eg Young Modulus). These models apply a network of masses and springs to simulate the mechanical behavior of deformable objects which is governed by spring constants. In general, these methods are not accurate in an absolute sense. However, many discrete approaches are fully adapted for interactive simulations.

Mass-spring models are probably the simplest and easiest models to implement among all discrete deformable models one can find in the literature (MEIER et al., 2005; NEALEN et al., 2006). In many interactive applications the MSM are used to simulate the behavior of different real-world objects. The MSM can be applied where the simulation based on continuous would be computationally unfeasible for real-time applications(JARAMILLO; PRIETO; BOULANGER, 2013).

There are three important advantages that make the MSM very atractive: simplicity in its mathematical formulation, great versatility for topological changes, and a structure well-suited for parallel computing. Hence, MSM techniques have been used to model deformable objects (NEALEN et al., 2006), for woven cloth simulation simulation (VOLINO; THALMANN, 2000) and soft organic tissues, like muscles, face or abdomen in virtual surgery applications (CHEN et al., 2007; ZERBATO; GALVAN; FIORINI, 2007; BASAFA; FARAHMAND; VOSSOUGHI, 2008; VICENTE et al., 2009; OTAMENDI, 2011) .

Regardless of the advantages and capabilities of the MSM for interactive mechanical simulations, it has drawbacks. The main limitation of the MSM is the difficulty of designing it to represent the mechanical behavior of deformable objects with enough accuracy. One way often used to surpass this limitation is to derive the MSM parameters, in particular the spring coefficients, from a continuous model due to its accuracy (BIANCHI et al., 2004; LLOYD; SZéKELY; HARDERS, 2007; BAUDET et al., 2009; NATSUPAKPONG; ÇAVUSOGLU, 2010; SAN-VICENTE; AGUINAGA; CELIGUETA, 2012; SILVA; GIRALDI; APOLINARIO, 2015).

Due to these facts, we have to solve a research problem which can be summarized as follows: on one hand we have interactive mechanical simulations and a model that for its characteristics has a vocation for real time application but is limited in accuracy. On the other hand, the continuous model that is accurate has been used to derive spring stiffnesses from elastic properties (Young's modulus and Poisson's ratio).

In this way, our work contributes to the parameterization of the MSM using a continuous model as reference. For this, in this chapter we present the theoretical basis for the mass spring model and pointed out the main solution strategies to the problem of parameterization.

3.1 MASS SPRING MODEL

One possibility for deformable objects simulation is to apply discrete models based on mass-spring models (MSM). Likewise in the FEM case, the object geometry is represented by a mesh with nodes $\mathbf{x}_i$, $i = 1, 2, \ldots, n$, which are connected according to some topology T. But, in the MSM technique, the mesh nodes are treated like mass points while each edge acts like a spring connecting two adjacent nodes. It is not used the notion of element and each mesh node $\mathbf{x}_i$ is represented in the global reference system as $\mathbf{x}_i = (x_{i,1}, x_{i,2}, x_{i,3})$. In order to compute the dynamic behavior of objects we derive forces at mass points from potential energies that preserve distances between masses.

The elastic potential energy of the whole body is:

$$E\left(\widehat{\mathbf{x}}\right) = \frac{1}{2} \sum_{(i,j)\in T} k_{ij}^{s} \left(\|\mathbf{l}_{i,j}\| - l_{i,j}^{0}\right)^{2}, \tag{3.1}$$

where $\widehat{\mathbf{x}} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ is the vector of all the particle positions like in section 2.2, $\mathbf{l}_{i,j} = \mathbf{x}_i - \mathbf{x}_j$, $l_{i,j}^0$ is the rest length of the spring with end points $\mathbf{x}_i$, $\mathbf{x}_j$, denoted by $i \to j$, and k_{ij}^s is the stiffness constant of the spring. Therefore, the overall elastic force at a mass point m_i is derived as:

$$\mathbf{f}_{elastic}^{i} = -\frac{\partial E}{\partial \mathbf{x}_i} = -\sum_{j\in V_i} k_{ij}^{s} \left(\|\mathbf{l}_{i,j}\| - l_{i,j}^{0}\right) \frac{\mathbf{l}_{i,j}}{\|\mathbf{l}_{i,j}\|}, \tag{3.2}$$

where V_i is the set of nodes linked to $\mathbf{x}_i$.

From equation. (3.2) it's easy to see that $\mathbf{f}_{elastic}^{i} = -\mathbf{f}_{elastic}^{j}$. Besides, in the absence of any damping, this conservative force will cause the system to oscillate around the solution of $E\left(\mathbf{x}\right) = 0$, which is the undeformed configuration $\widehat{\mathbf{x}}_0$. It corresponds to the reference configuration Ω_0 of the FEM model (notice that expression (2.10) also vanishes when $\widehat{\mathbf{x}} = \widehat{\mathbf{x}}_0$). To add damping to linear spring forces defined by equation (3.2), we apply the expression:

$$\mathbf{f}^i_{damp} = -\sum_{j\in V_i} k^d_{ij} \frac{\mathbf{l}_{i,j}.\dot{\mathbf{l}}_{i,j}}{\|\mathbf{l}_{i,j}\|} \frac{\mathbf{l}_{i,j}}{\|\mathbf{l}_{i,j}\|}, \tag{3.3}$$

with $\dot{\mathbf{l}}_{i,j} = \dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j$ and k^d_{ij} being the damping coefficient associanted to the spring $i \to j$.

The external force at node i is any general force acting on the system, e.g, gravitational force:

$$\mathbf{f}^i_{ext} = \mathbf{f}^i_{grav} = m_i\mathbf{g}, \tag{3.4}$$

So, given a particle i with mass m_i and position vector $\mathbf{x}_i$, the Newton's Laws allow to get the following dynamic equation of motion (BOURGUIGNON; CANI, 2000; LLOYD et al., 2008):

$$m_i\ddot{\mathbf{x}}_i + \mathbf{f}^i_{elastic} + \mathbf{f}^i_{damp} = \mathbf{f}^i_{ext}, \quad i = 1, 2, \ldots, n. \tag{3.5}$$

For simplicity, the sum of elastic and damping forces in expression (3.5) is often called the internal forces acting at node i, and is denoted by:

$$\mathbf{f}^i_{int} = \mathbf{f}^i_{elastic} + \mathbf{f}^i_{damp}. \tag{3.6}$$

The obtained equations (3.5) need initial conditions composed by the initial configuration and velocity to assure existence and uniqueness of the solution. These conditions will be also the input to numerical approaches based on Leapfrog method for time integration of the ordinary differential equations. The Leapfrog integration is a second order method which receives the positions $\mathbf{x}_i(t)$, velocity $\dot{\mathbf{x}}_i(t)$, and the acceleration $\ddot{\mathbf{x}}_i(t)$ computed from the Equation 3.5 in order to calculate the new system state to timestep $t + \triangle t$ by (POZRIKIDIS, 2008)

$$\dot{\mathbf{x}}_i(t + \triangle t) = \dot{\mathbf{x}}_i(t) + \frac{\triangle t}{2}\ddot{\mathbf{x}}_i(t), \tag{3.7}$$

$$\mathbf{x}_i(t + \triangle t) = \mathbf{x}_i(t) + \triangle t\dot{\mathbf{x}}_i(t + \triangle t). \tag{3.8}$$

By similar reasoning to obtain the equation (2.9), we can compute the linearized MSM equations of motion and write them in matrix form as:

$$M_{MSM}\ddot{\widehat{\mathbf{u}}} + K_{MSM}\widehat{\mathbf{u}} = \mathbf{f}, \tag{3.9}$$

where $\widehat{\mathbf{u}} = \widehat{\mathbf{x}} - \widehat{\mathbf{x}}_0$, $M_{MSM} = diag\begin{bmatrix} m_1I & m_2I & \ldots & m_nI \end{bmatrix}$ is the block diagonal mass matrix, and the stiffness matrix K_{MSM} encodes a first order approximation of the elastic spring forces nearby the rest positions. More specifically, the matrix K_{MSM} is the Jacobian of the force vector (Equation 3.2) and the Hessian of the elastic potential energy (Equation 3.1) with respect to the particle positions, computed at the undeformed configuration $\widehat{\mathbf{x}}_0$. It can be written in block form, where each block is given by,

$$K_{MSM}^{i,j} = \frac{\partial^2 E\left(\widehat{\mathbf{x}}_0\right)}{\partial \mathbf{x}_i \partial \mathbf{x}_j} \tag{3.10}$$

which is a 2 × 2 matrix in 2D or 3 × 3 matrix in 3D.

Paying attention to the MSM formulation (Equation (3.5)), we can note the presence of the three basic components of a mechanical system (KELLY, 2011): inertia, measured by nodal masses values m_i; stiffness, measured by the stiffness constant of the spring k_{ij}^s; damping, measured by the damping coefficient k_{ij}^d. The mechanical behaviour of a system involves the transfer of its potential energy to kinetic energy and of kinetic energy to potential energy, alternately, and optionally some energy is dissipated when the system is damped. So, the parameters m_i, k_{ij}^s and k_{ij}^d determine the mechanical behaviour of system assuming key role in the accuracy and realism of the simulations. Therefore, it becomes paramount the designing of systematic procedures to assign parameters to the MSM to ensure both visual realism and physical accuracy. In the next section we will discuss methodologies proposed in the literature by addressing this issue.

3.2 PARAMETERS IDENTIFICATION

Many improvements in the MSM designing have been presented by several authors in order to become the deformable model simulations more accurate and realistic. Some of them have conducted research that aim to developing new methods for obtaining of spring coefficients (LLOYD; SZéKELY; HARDERS, 2007; SILVA; GIRALDI; APOLINARIO, 2015) whereas others have proposed modify the traditional structure of the MSM by including force-based constraints (CHOI et al., 2004; BAUDET et al., 2009) or by adding special hinge as angular or bending springs (PROVOT, 1996; BOURGUIGNON; CANI, 2000; GIRALDI; ORTIZ; JR, 2006). Among them there is a consensus that exist an acceptable approach for computing the nodal masses values as in (DEUSSEN; KOBBELT; TüCKE, 1995; BAUDET, 2006), however, the problem of selecting the appropriate spring and damping constants is still unsolved (NATSUPARKPONG, 2009; SAN-VICENTE; AGUINAGA; CELIGUETA, 2012).

According to Lloyd *et al* (LLOYD; SZéKELY; HARDERS, 2007), two categories of methods can be identified in the estimation of the parameters for MSM in order to guarantee an accurate and realistic behavior: analytical derivation methods and data-driven techiniques that for simplicity we will call analytical-driven and data-driven methods, respectively.. The first class is composed by those methods that try to obtain the values of mass, stiffness and damping ratio that reproduce a known property of the reference model through theoretical considerations (ZERBATO; GALVAN; FIORINI, 2007). These methods seek to derive parameters starting from some analytical knowledge of the material or model, such as the FEM. For instance, we can seek for a linearized MSM model that produces elements with a stiffness matrix similar to that from linear FEM. This reasoning strategy based on the FEM formulation was started by Van Gelder (GELDER, 1998) who initially derived an MSM, with topology given by a triangular mesh, equivalent to the FEM in the context of linear elasticity. Specifically, it derives a formula for computing the spring stiffness coefficient of an edge according to the geometry of the triangles incident upon that edge as well as material properties (Young's modulus). However, this approach did not show positive results and the conclusion was that, in general, there is no possible solution that matches FEM and MSM stiffness matrices.

Some years later, (LLOYD; SZéKELY; HARDERS, 2007) demonstrated that there is a particular case where both matrices are equal. This particular case occurs when using equilateral triangle finite element and Poisson's ratio equal to $^1/_3$. The approach results in explicit formulas for the MSM stiffness coefficients for triangle, rectangle, and tetrahedron meshes. An extension of this work is found in (LLOYD et al., 2008) which presents formulas to derive the dynamic MSM parameters (mass and damping) as well. Also, it was demonstrated in (BAUDET et al., 2009) that Van Gelder's approach is restricted to null Poisson's ratio. In this reference it is supposed a linear elastic, isotropic and homogeneous materials and spring coefficients are determined to correctly simulate shear, elongation (tensile) for these mechanical systems. Firstly, the method computes the associated Lagrangian which depends on variables related to the material response to the shearing/elongation stress. Next, expressions for the Lagrangian extremum are computed to compose a system of equations together with the measured mechanical characteristics definitions. The idea is to build a set of equations whose solutions give the spring coefficients as function of the mechanical characteristics.

On the other hand, MSM models can be derived from a continuum approach by interpreting local expressions energy or force terms computed by finite difference or FEM formulations. In this way, in (DELINGETTE, 2008) it is considered an isotropic membrane represented by a triangular mesh and modeled by FEM. It is evaluated the membrane energy necessary to deform a single triangle and demonstrated that it is a function of square edge variation and the angles of the undeformed triangle. The obtained expression is composed by two terms: the first one is interpreted as the energy of three tensile bi-quadratic springs while the second term can be seen as three angular bi-quadratic springs that prevent changes in vertex angles. This approach still requires Poisson's ratio equal to $^1/_3$ and its extension to 3D is not yet available. A similar philosophy is used in (ETZMUSS; GROSS; STRASSER, 2003) but in this case the membrane model is discretized using a quadrilateral mesh and finite difference approaches. Besides, this reference derives spring models using discrete versions of evolution equations instead of discretizing the stretching energy as performed in (DELINGETTE, 2008).

The second category (data-driven approaches) is composed by methods that use a minimization procedure to find the model that shows the closest behavior to that of the observed (or simulated) deformable object. To implement such solution we must specify some properties and/or constraints for the MSM and then to seek for the other ones by optimizing an objective function that measures the similarity between the configurations of both the MSM and the reference model. In these cases, it is common to use genetic algorithms (VOLLINGER et al., 2009) and simulated annealing (MORRIS; SALISBURY, 2007). All theses methods share the same basic principle: applying random values to different springs properties and correct the ones that induce the greatest error in order to minimize the discrepancies. These methods are well suited for solving complex problems involving non-linearity and can handle with discrete properties of the MSM configuration, like the mesh topology (VOLLINGER et al., 2009). The main disadvantage of the use of these methods is the need for long computation times.

In order to avoid the limitations pointed out for the aforementioned model-driven techniques, in (NATSUPAKPONG; ÇAVUSOGLU, 2010) an optimization strategy is proposed to derive stiffness parameters based on the minimization of the difference between the FEM and the linearized MSM stiffness matrices. The derivation is performed nearby the steady-state solution of the FEM model; that means, the MSM governing equation is linearized by using the first order approximation for the spring forces computed at the rest nodal positions given by the FEM steady-state. Then, the optimization problem is set up and its solution gives the stiffness coefficients of the springs. Another optimization approach can be obtained using the eigendecomposition of the solution space following a methodology imported from modal analysis in vibration theory. This approach performs the analysis of the spectral properties and principal directions of the stiffness matrices in order to find most suitable parameters for MSM (OTAMENDI, 2011).

3.2.1 Nodes Mass Identification

The computation of the mass matrix M_{FEM} given by equation (2.8) does not offer a systematic way to map the elements of matrix M_{FEM} to the MSM masses. Therefore, to identify the nodal masses of the MSM some heuristics are widely used.

The first possibility to derive the mass m_i is to consider only the diagonal elements of matrix M^e associated with the element Ω_e in expression (2.4), which allows an intuitive physical association given by

$$m_i = \widehat{m}_{ii}^e, \tag{3.11}$$

where $\widehat{m}_{ii}^e$ is the diagonal terms from the finite element lumped mass matrix (NATSUPAKPONG; ÇAVUSOGLU, 2010).

On the other hand, if the material density ρ is known, the node mass m_i can be computed considering the node i surrounding area/volume, so the problem rephrases in defining a surface/volumetric region belonging to each node. In this case, a general expression to compute m_i is given by (BAUDET, 2006; LLOYD et al., 2008; SALA et al., 2011)

$$m_i = \sum_{e \in \Omega_e} \rho \frac{\Lambda_e}{n_n}, \tag{3.12}$$

where Λ_e is area/volume from polygon/polyhedron, n_n are the number of nodes of the of polygon/polyhedron, Ω_e is the set of polygon/polyhedron sharing the node i.

3.2.2 Springs Stiffness Identification

As mentioned previously, the springs stiffness coefficients can be determined through parameter estimation methodologies, to fit the MSM model to an experimentally measured response or by mean of theoretical considerations about some analytical knowledge from material or model. It is already known that the stiffness coefficients can be analytically computed under certain conditions. For instance, the spring coefficient in the 4-node rectangle square element can be computed by expression

$$k_{edge} = \sum_{e \in \Omega_e} \frac{5}{16} tE, \quad k_{diagonal} = \frac{7}{16} tE, \tag{3.13}$$

where t is thickness and E is Young Modulus, since Poisson's ratio have been equals to 1/3 (LLOYD; SZéKELY; HARDERS, 2007).

On the other hand, in the approach proposed by (GELDER, 1998), the stiffness constant of a spring, with rest length c, representing the common edge of two neighboring triangles of the mesh by (T_i) of surfaces $|T_i|$ and with edges c, a_i, b_i (with $i \in \{1, 2\}$) is given by

$$k_c = \sum_{i=1}^{n} \frac{E}{1+\nu} \frac{|T_i|}{c^2} + \frac{E\nu}{1-\nu^2} \frac{a_i^2 + b_i^2 - c_i^2}{8|T_i|}, \tag{3.14}$$

with ν, the Poisson's ratio and E, the corresponding 2D Young's modulus of the simulated material. Moreover, Van Gelder's results are restricted to $\nu = 0$ to avoid negative value of k_c. However, these expressions are only defined for the 2D case. For the 3D case there is no general solution to compute analytically the springs stiffness coefficients as shown in the work (GELDER, 1998; LLOYD; SZéKELY; HARDERS, 2007; BAUDET et al., 2009). This limitation leads us to methods that use a minimization procedure to find the mass spring model that shows the closest behavior to that of the reference model or data experimentally obtained.

In the static equilibrium, the MSM equations can be written as (LLOYD; SZéKELY; HARDERS, 2007)

$$\mathbf{f}_{elastic}\left(\widehat{\mathbf{x}}, \boldsymbol{\phi}\right) = \mathbf{f}_{ext} \tag{3.15}$$

where $\mathbf{f}_{elastic}\left(\widehat{\mathbf{x}}, \boldsymbol{\phi}\right)$ is an $dn \times 1$ vector containing the spring forces, $\widehat{\mathbf{x}} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ is an $dn \times 1$ vector that contains all the d-dimensional vertex positions $\mathbf{x}_i$, and $\boldsymbol{\phi} = (k_{12}, k_{13}, \ldots)$ is the vector of the unknown spring stiffness coefficients. The vector $\mathbf{f}_{ext}$ on the right hand side contains the external forces at each node. Given the external forces $\mathbf{f}^{ref}$ and the corresponding vertex positions $\mathbf{x}^{ref}$ of a reference model in the deformed state, the goal is to optimize the spring coefficients so that the MSM and the reference model behave similarly. The similarity criterion is usually defined as the Euclidean distance between the nodes of the MSM and the reference model often expressed by

$$\varphi\left(\boldsymbol{\phi}\right) = \sum_{i} \left\| \mathbf{x}_i^{ref} - \mathbf{x}_i\left(\boldsymbol{\phi}, \mathbf{f}^{ref}\right) \right\|^2, \tag{3.16}$$

where the positions $\mathbf{x}_i\left(\boldsymbol{\phi}, \mathbf{f}^{ref}\right)$ are the equilibrium positions for a specific parameter vector $\boldsymbol{\phi}$ and external forces $\mathbf{f}^{ref}$. The functional $\varphi(\boldsymbol{\phi})$ must be minimized to obtain optimal parameters $\bar{\boldsymbol{\phi}} = \arg\min\limits_{\boldsymbol{\phi}} \varphi(\boldsymbol{\phi})$. This involves computing the equilibrium positions for each candidate parameter vector $\boldsymbol{\phi}$. In this optimization process, classical methods can be used. However, usually they get trapped in local minima. Therefore, mostly alternative methods have been used as genetic algorithms (VOLLINGER et al., 2009) and simulated annealing (MORRIS; SALISBURY, 2007). Following the static equilibrium reasoning we can develop an objective function that measures the similarity between the reference model and the linearized MSM stiffness matrices. Thereby, equation (3.16) is replaced by

$$\bar{\boldsymbol{\phi}} = \arg\min_{\boldsymbol{\phi}} \left\| \mathbf{K}^{ref} - \mathbf{K}_{MSM} \right\|^2 . \tag{3.17}$$

This methodology is followed by (NATSUPAKPONG; ÇAVUSOGLU, 2010; OTAMENDI, 2011) which determines stiffness parameters of linear mass-spring models with the linear FEM model as the reference one.

Assumption 16. *Henceforth, in the remainder of this thesis, we call "Matrix Approach" the approaches that use the expression (3.17).*

In the next section we present an overview of the simulated annealing method which will be applied in the solving of the optimization problems of this thesis.

3.2.3 Simulated Annealing

The idea of Simulated Annealing (SA) comes from a paper published by Metropolis (METROPOLIS et al., 1953) that was generalized by the Kirkpatrick *et al* (KIRKPATRICK; GELATT; VECCHI, 1983) to include a temperature schedule for efficient searching. In this latter, the SA is presented as probabilistic method for finding the global minimum of a cost function that may possess several local minima (BERTSIMAS; TSITSIKLIS, 1993).

The SA algorithm was inspired from the physical annealing analogy. Physical annealing is a process in which a solid is first heated until all particles are randomly arranged in a liquid state, followed by a slow cooling process. At each (cooling) temperature enough time is spent for the solid to reach thermal equilibrium, where energy levels follow Boltzmann distribution. As temperature decreases the probability tends to concentrate on low energy states.

In simulated annealing we keep a temperature variable to simulate this heating process. We initially set it high and then allow it to slowly 'cool' as the algorithm runs. While this temperature variable is high the algorithm will be allowed, with more frequency, to accept solutions that are worse than our current solution. This gives the algorithm the ability to jump out of any local optimums it finds itself in early on in execution. As the temperature is reduced so is the chance of accepting worse solutions, therefore allowing the algorithm to gradually focus in on a area of the search space in which hopefully, a close to optimum solution can be found. This gradual 'cooling' process is what makes the simulated annealing algorithm remarkably effective at finding a close to optimum solution when dealing with large problems which contain numerous local optimums. The Algorithm 3.1 shows the typical steps carried out during the optimization process with simulated annealing.

Algorithm 3.1 A General Simulated Annealing Algorithm

```
Initialization(Current_Solution, Temperature)
Calculation of the Current_Cost
LOOP
  New_State
  Calculation of the New_Cost
  IF $\triangle$ (Current_Cost $-$ New_Cost) $\leq 0$ THEN
    Current_State = New_State
  ELSE
    IF $\exp\left(\frac{\text{Current\_Cost}-\text{New\_Cost}}{\text{Temperature}}\right) > \text{Random}(0,1)$ THEN
      Current_State = New_State
    ELSE
      Reject
    ENDIF
  ENDIF
  Decrease the temperature
EXIT When STOP_CRITERION
ENDLOOP
```

3.3 CONCLUSIONS

The MSM is simple, fast and suitable for interactive mechanical simulations. In order to ensure accuracy and realism in the simulation, the MSM parameters (nodal mass, stiffness constant of the spring and damping constant) that control the mechanical behaviour must be assigned rightly. Parameterization methods have been proposed to address this issue.

Some parameterization methods use analytical expressions to calculate the parameters and so are simple and fast. However, they impose restrictions such as: specific values of physical constants (Young Modulus, Poisson's ratio); linearization of elastic force; specific continuous model and its structures. Others parameterization methods compute the parameters solving an optimization problem that fitting of deformation of the MSM to some reference data. These methods are not subject to any condition other than the reference data. The disadvantage of them is the necessity for long computation times.

In the next chapters, we will discuss a few more about the advantages and disadvantages of the strategies for parameterization of the MSM. From these discussion, we will propose two methods to the MSM parameterization: a) *The acceleration-based approach* that proposes the parameterization of the MSM using FEM as reference model in an environment free of static equilibrium and linearization of elastic force; b) *The IGA-based approach* that performs the MSM parameterization using NURBS-based IGA as reference model with higher order elements to improve accuracy and realism.

These methods will described in the next chapters. The acceleration-based approach will be presented in Chapter 4. The IGA-based approach will be presented in Chapter 5.

Chapter

4

In general, the approaches to the MSM parameterization are built on static equilibrium scenario and linearized elastic force. Besides, restrictions on physical constants (Young Modulus and Poisson's ratios) and on reference model need to be applied to enable the parameterization. In this chapter we propose a method that overcomes these restrictions without interfering with any of the good qualities of the MSM.

PARAMETERIZATION BASED ON FEM

The problem of the MSM parameterization is an issue still investigated nowadays. In particular, the selection of proper values for the spring and damping constants calculated without manual tuning is an open problem as mentioned in section 3.2. The main argument for this claim is that the model parameters (spring coefficients, damping constants and masses) are not related to elastic material constitutive laws in an obvious way (LLOYD et al., 2008) and there is no general physically based or systematic method in the literature to determine the element types or parameters from physical data or known constitutive behavior (NATSUPAKPONG; ÇAVUSOGLU, 2010).

To tackle this challenge the researchers try to incorporate the accuracy and realism from continuous models into the MSM in two ways: a) developing analytical expressions linking properties of continuous model with the MSM (analytical-driven methods); b) fitting of deformation of the MSM to some reference data by modifying its inner parameters (data-driven approaches), where the reference data may be obtained experimentally or generated by a continuous model like FEM (see section 2.2) or IGA (see section 2.5). Of course, there are advantagens and disadvantagens to both approaches.

The analytical-driven methods are, in general, fast and able to give good results under certain conditions. About these restrictions, we may recall that the analytical expressions to calculate spring coefficients require Poisson's ratio be assigned to a fixed value ,(VAN-GELDER; WILHELMS, 1997; ETZMUSS; GROSS; STRASSER, 2003), restrictions to angles between the springs (BAUDET et al., 2009), and linearization of MSM when the derivation of analytical expression occurs from the stiffness matrices of models (LLOYD; SZéKELY; HARDERS, 2007). For the latter, the analytical expression is only valid for a stiffness matrix built with a linear finite element (mandatorily triangle or tetrahedron) and Poisson's ratio must assume a predefined value, generally $^1/_3$.

The data-driven approaches is not subject to any condition other than the reference data. It does not need any previous detail or information about mechanical behaviour of the body apart from the reference data. The body can show behaviour linear or nonlinear, the derivation process will compute the parameters best-suited for the specific reference data. Therefore, these methods generally lead to optimization problem which has as main disadvantage the long computation time to find the optimal set of parameters.

In an overview about both approaches, we can affirm that:

- they pursued a similarity relationship from the continuum knowledge which can be continuum mechanics or a continuum-based model as FEM;
- although there are approaches based on dynamic equilibrium both have mainly focused on the static equilibrium in which estimate the stiffness coefficients is the central goal;
- when procedure to estimation are built on FEM, both approaches rely exclusively on linear elements;
- data-driven and analytical-driven methods to estimate spring coefficients by using the matrix approach depend on linearization of elastic force;
- and finally, reviewing literature of the MSM parameterization, we find out that to FEM models with higher order polynomial interpolation function have not been considered in the MSM derivation literature, at least until current date.

These considerations lead us to research and propose a methodology that was free of:

- Static equilibrium;
- Linearization of elastic force;
- Restrictions of any order in Poisson's ratio value;
- The use of the FEM as reference model with linear elements.

Aside from the arguments cited above in favor of the derivation from FEM with higher order elements we also made efforts to contribute to an approach that would improve the following limitations: conditioning for validity of the estimated parameter as those imposed by analytical-driven methods; dependency of linear elements (both data-driven and analytical-driven methods); linearization of MSM and necessity of static equilibrium.

Since many approaches accomplish the parameters derivation of MSM from a continuum-based model, we also start our work research to find out the set of parameters that makes the MSM behaves similarly to FEM. Thus, we propose the parametrization based on acceleration approach. In this method, the main goal is achieve the parameterization of the MSM in an environment free of static equilibrium and linearization of elastic force. For this, our proposal is to define a new method to design the mass-spring systems that uses a data-driven strategy under a dynamic scenario. The method may be summarized into two stages: (a) Simulate the deformable object using a reference model and keep the system state (position and velocities) of the particles; (b) Solve an optimization problem based on the acceleration of the reference and MSM models in order to compute the stiffness parameters.

The derivation of the MSM parameters based on accelelation strategy was published in (SILVA; GIRALDI; APOLINARIO, 2015) under the title "A new optimization approach for mass-spring models parameterization", in Elsevier's journal Graphical Models. Our proposal for MSM parameters derivation is based on the independence of the static equilibrium and elastic force linearization.

4.1 ACCELERATION COST FUNCTION

To achieve independence from static equilibrium strategy and the linearization of elastic force we start with the motion equation of the MSM (see Equation (3.5)) in order to define an objective function that expresses the optimization problem whose the solution will result in estimated stiffness coefficients. We shall observe that the MSM motion equation (3.5) can be rewritten to isolate the particle acceleration $\ddot{\mathbf{x}}_i(t)$:

$$\ddot{\mathbf{x}}_i(t) = \frac{\mathbf{f}^i_{ext}(t) - \mathbf{f}^i_{int}\left(\mathbf{x}_i(t), \boldsymbol{\phi}\right)}{m_i} \equiv \ddot{\mathbf{x}}^{msm}_{i,t}, \tag{4.1}$$

where $\mathbf{f}^i_{int}\left(\mathbf{x}_i(t), \boldsymbol{\phi}\right)$ is the vector in expression (3.6) containing the internal forces that depend on the stiffness and damping parameters assembled in the array ϕ. In the optimization problem to be considered, we do not include the damping effects; so, the vector $\mathbf{f}^i_{int}\left(\mathbf{x}_i(t), \boldsymbol{\phi}\right)$ includes spring forces only. Consequently, given position $\mathbf{x}_i\left(t\right)$, velocity $\dot{\mathbf{x}}_i\left(t\right)$ computed from the reference model, masses m_i calculated by the equation (3.11) or (3.12), we shall observe that $\ddot{\mathbf{x}}^{msm}_{i,t} = \ddot{\mathbf{x}}^{msm}_{i,t}\left(k_1, k_2, \ldots, k_m\right)$ in expression (4.1). Therefore, we can define the cost function φ as

$$\varphi\left(\boldsymbol{\phi}\right) = \sum_i \sum_{t=0}^{\tau} \left\| \ddot{\mathbf{x}}^{ref}_{i,t} - \ddot{\mathbf{x}}^{msm}_{i,t} \right\|^2, \tag{4.2}$$

where $\boldsymbol{\phi} = \left(k_1, k_2, \ldots, k_m\right)^T$ is the vector containing stiffness coefficients and τ is the last instant of simulation. So, we should solve the optimization problem

$$\bar{\boldsymbol{\phi}} = \arg\min_{\phi} \varphi\left(\boldsymbol{\phi}\right), \tag{4.3}$$

in order to find the best values for the parameters in the vector $\boldsymbol{\phi}$.

To avoid that the optimization occurs at the whole solution space of the reference model which would lead to computation cost prohibitively large, we impose a strong simplification by observing that, to find $\bar{\boldsymbol{\phi}}$ in expression (4.3) we need to solve the equations $\frac{\partial \varphi}{\partial k_\alpha} = 0$, $\alpha = 1, \ldots, m$, given by (SILVA; GIRALDI; APOLINARIO, 2015):

$$\sum_i \sum_{t=0}^{\tau} \left(\ddot{\mathbf{x}}^{ref}_{i,t} - \ddot{\mathbf{x}}^{msm}_{i,t} \right) \cdot \frac{\partial}{\partial k_\alpha} \ddot{\mathbf{x}}^{msm}_{i,t} = 0. \tag{4.4}$$

The key idea is to find out the dominant terms in expression (4.4) and then to compute k_α that minimize them. So, let us take the configuration with highest elastic FEM energy in the simulation times $t_0, t_1, \ldots, \tau$.

If the FEM elastic energy is maximum at $t = \bar{t}$ we expect that the internal forces, and consequently the particle accelerations, get also higher values at this simulation time. Therefore, we could postulate that the dominant term in expression (4.4) is the one corresponding to $t = \bar{t}$ and, consequently, a guess for the desired solution $\bar{\boldsymbol{\phi}}$ would be:

$$\bar{\bar{\boldsymbol{\phi}}} = \arg\min_{\phi} \sum_{i} \left\| \left(\ddot{\mathbf{x}}_{i,\bar{t}}^{ref} - \ddot{\mathbf{x}}_{i,\bar{t}}^{msm} \right) \right\| . \tag{4.5}$$

4.2 GROUPING SPRING

We can rewrite the elastic force in the expression (3.2) as

$$\mathbf{f}_{elastic}^{i} = \sum_{j \in V_i} \left[k_{i,j}^{s} \mathbf{w}_e \left(\mathbf{x}_i, \mathbf{x}_j \right) \right] \tag{4.6}$$

where

$$\mathbf{w}_e \left(\mathbf{x}_i, \mathbf{x}_j \right) = \begin{pmatrix} (\mathbf{w}_e^{ij})_1 \\ (\mathbf{w}_e^{ij})_2 \\ (\mathbf{w}_e^{ij})_3 \end{pmatrix} = -\left(\|\mathbf{l}_{i,j}\| - l_{i,j}^{0} \right) \frac{\mathbf{l}_{i,j}}{\|\mathbf{l}_{i,j}\|}.$$

For simplicity, let us assume that the cardinality of V_i is a constant N, for all i, then

$$\mathbf{f}_{elastic}^{i} = \begin{pmatrix} (\mathbf{w}_e^{i1})_1 & (\mathbf{w}_e^{i2})_1 & (\mathbf{w}_e^{i3})_1 & \cdots & \left(\mathbf{w}_e^{iN}\right)_1 \\ (\mathbf{w}_e^{i1})_2 & (\mathbf{w}_e^{i2})_2 & (\mathbf{w}_e^{i3})_2 & \cdots & \left(\mathbf{w}_e^{iN}\right)_2 \\ (\mathbf{w}_e^{i1})_3 & (\mathbf{w}_e^{i2})_3 & (\mathbf{w}_e^{i3})_3 & \cdots & \left(\mathbf{w}_e^{iN}\right)_3 \end{pmatrix} \begin{pmatrix} k_{i,1} \\ k_{i,2} \\ k_{i,3} \\ \vdots \\ k_{i,N} \end{pmatrix}, \tag{4.7}$$

and hence

$$\mathbf{f}_{elastic} = \begin{pmatrix} W^1 & & & & \\ & W^2 & & & \\ & & \cdots & & \\ & & & W^{n-1} & \\ & & & & W^n \end{pmatrix} \begin{pmatrix} \phi^1 \\ \phi^2 \\ \vdots \\ \phi^{n-1} \\ \phi^n \end{pmatrix} = A_{MSM} \boldsymbol{\phi}, \tag{4.8}$$

where $\mathbf{f}_{elastic} \in \mathbb{R}^{3n}$, $W^i \in \mathbb{R}^{3 \times N}$and $\phi^i = \begin{pmatrix} k_{i,1} & k_{i,2} & k_{i,3} & \ldots & k_{i,N} \end{pmatrix}^T \in \mathbb{R}^N$. Therefore, $A_{MSM} \in \mathbb{R}^{3n \times Nn}$and $\boldsymbol{\phi} \in \mathbb{R}^{Nn}$.

Dropping the damping term of the MSM motion equation (3.5) , we have

$$m_i \ddot{\mathbf{x}}_i + \mathbf{f}_{elastic}^{i} = \mathbf{f}_{ext}^{i}, \quad i = 1, 2, \ldots, n, \tag{4.9}$$

is equivalent to

$$M_{MSM} \ddot{\hat{\mathbf{x}}} + A_{MSM} \boldsymbol{\phi} = \mathbf{f}_{ext},$$

where $\widehat{\mathbf{x}} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ and $M_{MSM} = diag \left[\begin{array}{cccc} m_1 I & m_2 I & \ldots & m_n I \end{array} \right] \in \mathbb{R}^{3n \times 3n}$.

Therefore, internal forces $\mathbf{f}^i_{int}$ and external forces $\mathbf{f}^i_{ext}$ in expression (4.9) can be assembled in a global expression:

$$A_{MSM}\boldsymbol{\phi} = \mathbf{b}, \tag{4.10}$$

where $\mathbf{b} = \mathbf{f}_{ext} - M_{MSM}\ddot{\widehat{\mathbf{x}}}$.

Thus, equation (4.10) can be treated as a linear system with Nn unknowns and $3n$ equations, where often $Nn > 3n$, ie, undetermined system with (in general) more than one solution. To deal with such ill-posed problem we could impose a set of constraints involving the stiffness coefficients in order to get a determined or an overdetermined system. However, in this scenario, we should be carefull about the effects of these constraints in the material properties; particularly for homogeneity and isotropy. In the MSM model, these properties are consequences of the mesh topology, the stiffness coefficients and rest lenghts, which control the degrees of freedom of the system. However, the larger the number of degrees of freedom is the more difficult to predict the variations in the mentioned properties, which may bring undesirable consequences in the MSM derivation approach. So, our strategy is steered by simplification: (a) We represent the reference configuration Ω_0 through a structured mesh; (b) The length l^0_{ij} is the length of the edge (i, j) in the reference configuration; (c) Springs with the same rest length have the same stiffness coefficient. In this way, we can predict, at least qualitatively, the consequences of the MSM setup in the corresponding material properties.

Due to the assumptions (a)-(c) it is convenient to group the $N.n$ springs by the rest lengths l^0_1, l^0_2,...,$l^0_{\bar{m}}$, because we have, in general, that $\bar{m} \ll N.n$. Formally, this process is equivalent to arrange each linear expression in equation (4.7) as:

$$\begin{aligned} \left(\mathbf{w}^{i1}_e\right)_j k_{i,1} + \left(\mathbf{w}^{i2}_e\right)_j k_{i,2} + \left(\mathbf{w}^{i3}_e\right)_j k_{i,3} + \ldots \\ \ldots + \left(\mathbf{w}^{iN}_e\right)_j k_{i,N} = k_1 A^i_{j,1} + k_2 A^i_{j,2} + \ldots + k_{\bar{m}} A^i_{j,\bar{m}}, \end{aligned} \tag{4.11}$$

where:

$$A^i_{j,\alpha} = \sum_{s=1}^{N} \left(\mathbf{w}^{is}_e\right)_j f_\alpha\left(k_{i,s}\right), \tag{4.12}$$

$$f_\alpha(k_{i,s}) = \begin{cases} 1 & if\ l^0_{i,s} \in [l^0_\alpha(1-\gamma), l^0_\alpha(1+\gamma)] \\ 0 & otherwise \end{cases}, \quad \alpha = 1, \ldots, \bar{m}, \tag{4.13}$$

with $0 < \gamma < 1$ being a parameter to control the number of spring groups. This factor γ controls the granularity of the spring groups around the rest length l^0_α. If it increases then the granularity decreases, therefore we will have few spring groups. Otherwise, if it decreases then the granularity increases, which implies an increase in the number of spring groups.

In this way, we can rewrite the linear system (4.10) such that, $\bar{m}$ becomes the new number of unknowns:

$$\bar{\mathbf{A}}\bar{\boldsymbol{\phi}} = \mathbf{b}, \tag{4.14}$$

where the $3n \times \bar{m}$ matrix $\bar{\mathbf{A}}$ encode spring force coefficients grouped by rest length, and $\bar{\boldsymbol{\phi}} = (k_1, k_2, \ldots, k_{\bar{m}})^T$is the array of stiffness constants to be computed.

4.3 DEFORMATION MEASUREMENT

In some situations, including the parameterization of the MSM, it is important to measure the strain somehow. Therefore, in the following is presented a simple measure of strain that will be used in the rest of this thesis.

Strain represented by the Greek letter ϵ, is a term used to measure the deformation or extension of a body that is subjected to a force or set of forces. The strain of a body is defined as the change in length divided by the initial length. In the simplest case (one-dimensional), an edge (that represents a bar or a spring) can grow from its original length l_0 to its final length l. The change in length $l - l_0$ represents the deformation of the edge. In the literature (SADD, 2009; VABLE, 2009), the normal or extensional strain is the intensity of deformation defined by the change in length divided by the initial length. the strain can be defined as the ratio ϵ between the amount of stretching and the original length given by

$$\epsilon = \frac{l - l_0}{l_0}. \tag{4.15}$$

To compute the global deformation for a mesh with n edges that have undergone deformation, we can simplify the computation by using the expression

$$\epsilon_G = \frac{1}{n}\left(\sum_{i=1}^{n} \left|\epsilon\left(l^i, l_0^i\right)\right|\right), \tag{4.16}$$

where ϵ_G is is the average deformation which is a result of summation involving individual deformation $\epsilon\left(l^i, l_0^i\right)$ of each edge i.

Furthermore, the global deformation express in (4.16) can be interpreted by taking the relative percentage defined as

$$\%\epsilon_G = \epsilon_G.100. \tag{4.17}$$

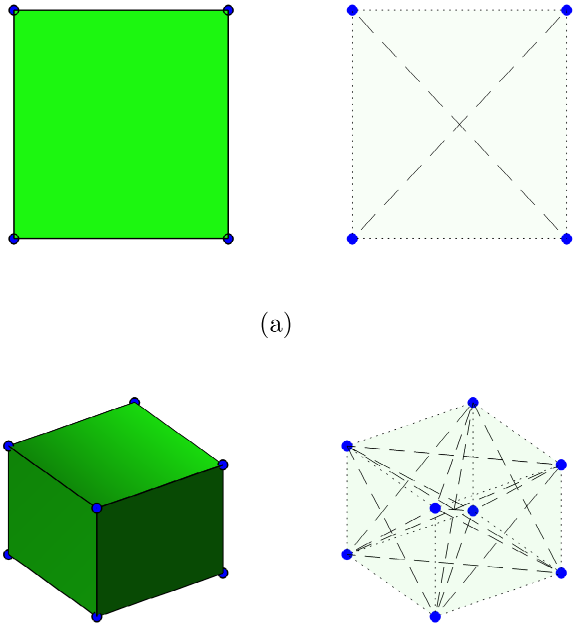
4.4 PROPOSED METHOD

As seen in previous sections, we resort to the use a cost function based on the acceleration of both models to ensure that our approach does not rely on static equilibrium neither of the linearization of elastic force. The acceleration is a measure that involves the time (it is the second derivative of the position with respect to time). So, we need temporal evolution of the reference model to solve the optimization problem once the cost function has a time component.

To prevent a optimization process at the whole solution space (temporal evolution of reference model) which would lead to computation cost prohibitively large, we made a strong simplification which is find out the dominant term that occurs at $t = \bar{t}$ where the FEM elastic energy is maximum. Furthermore, aiming to control the number of variables to be calculated on the optimization, we can optionally accomplish a grouping of springs by their rest length.

That said, the proposed method is summarized by the following procedures:

1. Initialization: (a) Define material properties (Young's module and Poisson's ratio), mesh topology and numerical characteristics for the reference model; (b) Set the MSM geometry as the same of the reference model and compute the masses using expression (3.12); c) Set the initial conditions for the reference model. For MSM meshing from FEM we use the mapping shown in Figure 4.1.



(a)

(b)

Figure 4.1: Mapping FEM to MSM meshes for 2D (a) and 3D (b) cases. On the left of each sub-figure is the FEM mesh (blue points connected by black line). On the right of each sub-figure is the MSM mesh which contains nodal masses linked by stretch (dotted black) and shear (dashed black) springs.

2. Simulate the reference model for $t = t_0, t_1, \ldots, \tau$ without damping and external force.

3. Keep the vectors $\left(\widehat{\mathbf{x}}_{\bar{t}}^{ref}, \dot{\mathbf{x}}_{\bar{t}}^{ref}, \ddot{\mathbf{x}}_{\bar{t}}^{ref}\right)$ at time $t = \bar{t}$ where the elastic energy is maximum.

4. Compute $\ddot{\mathbf{x}}_{i,t}^{msm} = \ddot{\mathbf{x}}_{i,t}^{msm}\left(k_1, k_2, \ldots, k_m\right)$ or optionally $\ddot{\mathbf{x}}_{i,t}^{msm} = \ddot{\mathbf{x}}_{i,t}^{msm}\left(k_1, k_2, \ldots, k_{\bar{m}}\right)$, by inserting $\left(\widehat{\mathbf{x}}_{\bar{t}}^{ref}, \dot{\mathbf{x}}_{\bar{t}}^{ref}, \ddot{\mathbf{x}}_{\bar{t}}^{ref}\right)$ in equation (4.1).

5. Calculate $\bar{\bar{\boldsymbol{\phi}}}$ by solving the optimization problem (4.5).

6. Calculate damping using expression

$$c_{ij} = \alpha^{-1}\left(\frac{2\sqrt{k_{ij}\left(m_i + m_j\right)}}{l_{ij}^{0}}\right), \tag{4.18}$$

   where $\alpha \in \mathbb{N}^{+}$ is scale factor to control the system damping effect.

From the mathematical viewpoint, the optimization problem in expression (4.5) can be solved by traditional least square techniques. However, numerical and computational aspects must be also considered. In our implementation, we use the simulated annealing, implemented in our Multi-model Analysis Framework (described in section 3.2.3). As usual practice, the criteria for stopping is a certain number of iterations (or temperatures) has passed without acceptance of a new solution. However, the obtained solution can give arbitrary stiffness coefficients such as negative or very small values if restrictions are not enforced. Negative spring stiffness parameters can be avoided by limiting the search space to non-negative values (LLOYD; SZéKELY; HARDERS, 2007). Also, to prevent higher values that generate unstable results, we propose to limit the search space by expression

$$k_b^{ij} = \sum_{e \in \Omega_e} \sqrt{d}\,\frac{E\left(1 + \nu\right)\max\left(\mathit{\Lambda}_e\right)}{2\left(l_{ij}^{0}\right)^{2}}, \tag{4.19}$$

where $\mathit{\Lambda}_e$ is area/volume from polygon/polyhedron element, E is Young Modulus, ν the Poisson's ratio, d is problem dimension 2 or 3, l_{ij}^{0} is rest length, and Ω_e is the set of polygon/polyhedron sharing the node i.

4.5 RESULTS

The validity of the parameterization based on acceleration approach happened through the simulation of 2D and 3D elastic objects. The first experiment was conducted using the MSM itself as a reference model that aimed to assess the method's behavior with a non-linear model. Together with the first experiment we analyse the influence of α (expression (4.18)) in the time step $\triangle t$ with simulations that demonstrate the damping effect in the system . In the sequel, a set of simulations and tests were conducted with the linear elastic FEM as reference model: a) comparison of our proposal against the matrix approach which was developed by Surya (NATSUPAKPONG; ÇAVUSOGLU, 2010) and Otamendi (OTAMENDI, 2011); b) the effects of the geometry discretization on our approach; and c) the sensitivity of the solution obtained by our method to the Poisson's ratio value. We summarize the experiments with its objects and their geometries in Table 4.1.

Experiment	Objects	Size(meter)	Nodes	Springs
I)	Square membrane	2×2	9×9	272
II)	Square membrane	2×2	5×5	72
III)	Square membrane	2×2	9×9	72
	Elastic cube	$2 \times 2 \times 2$	$7 \times 7 \times 7$	3258
	Elastic plate	$2 \times 2 \times 0.4$	$7 \times 7 \times 2$	673
IV)	Square membrane	2×2	101	328
	Circular membrane	Radius=1	89	316
V)	Square membrane	2×2	9×9	272

Table 4.1: Experiments overview: I) Derivation with non-linear model; II) Analyzing the influence of α and damping effect; III) Performance comparative against the matrix approach; IV) Effects of the geometry discretization; V) Sensitivity to the Poisson's ratio value.

In order to provide quantitative results, the percentage of root mean square error $\%e_{rms}$, and the percentage of maximum error $\%e_{max}$ of Euclidean distance between reference model and MSM nodes are calculated by (NATSUPAKPONG; ÇAVUSOGLU, 2010) :

$$\%e_{rms} = \frac{e_{rms}}{o_{max}} * 100, \tag{4.20}$$

$$\%e_{max} = \frac{e_{max}}{o_{max}} * 100, \tag{4.21}$$

where $e_{rms} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_i^{REF} - x_i^{MSM}\right)^2}$, $e_{max} = \max\left|x_i^{REF} - x_i^{MSM}\right|$, o_{max} is the maximum Euclidean distance between the undeformed and the deformed configurations of reference model, n is the number of nodes in the models, and x_i^{REF} and x_i^{MSM} denote the positions of corresponding nodes of the reference and MSM models.

Experiment I) *Derivation with a non-linear model.* We instantiate a MSM model by setting up the masses $m_i = 2kg$, the stiffness $k_{edge} = 500$, $k_{diag} = 400$, damping coefficients null, mesh geometry given by Figure 4.2 with the rest length assigned to $l^0_{edge} = 0.250m$, $l^0_{diag} = 0.353m$. For time integration we use the Leapfrog method with time step $\triangle t = 1/120$ as the numerical integration scheme (expressions (3.7-3.8). Then, we execute the our method's procedures (just steps 1-5). For $t = \overline{t}$, the obtained MSM parameters are: $k_{edge} = 459.130$, $k_{diag} = 406.514$.

In order to quantify the quality of the solution, we consider the already stated setup but we apply an external force in the positive x-direction and negative y-direction: $\mathbf{f}\left(f_x, f_y\right) = (5, -10)$ which in static equilibrium configuration yields a strain around 10% that is computed by using expression (4.17). We solve the static equilibrium equations and compute the percentage of root mean square error $\%e_{rms}$, and the percentage of

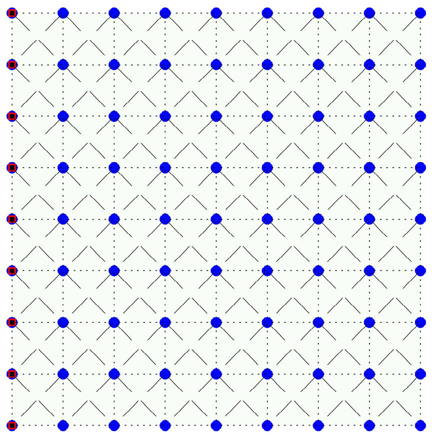
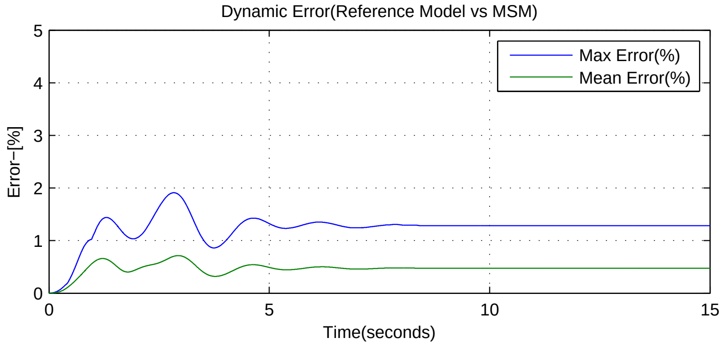


Figure 4.2: The MSM mesh for square elastic membrane. Fixed nodes are colored in red darkness and free nodes are colored as blue. The nodes are linked by stretch (dotted black) and shear (dashed black) springs

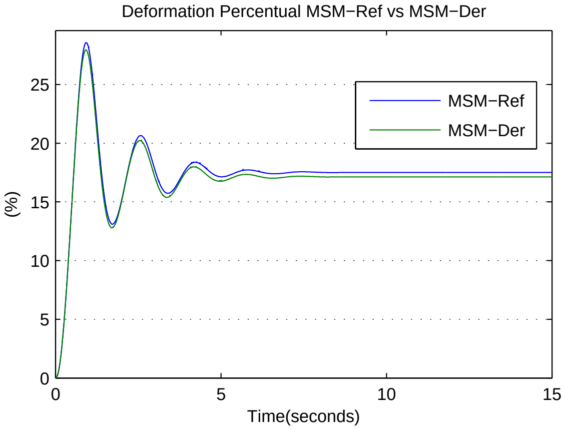
maximum error $\%e_{max}$ of Euclidean distance between reference model and MSM nodes. Here, we observe errors of $\%e_{rms} = 0.69$, $\%e_{max} = 1.20$ between the configuration of both MSM and reference model. Therefore, we conclude that the derived MSM and the reference model behave very similar.

A computational experiment also carried out to study the evolution of the MSM obtained over time by using the Leapfrog method and time step $\triangle t = {}^{1}/_{480}$. For this, we set the MSM with obtained parameters k_{edge}, k_{diag} and we apply a constant force in the positive x-direction and negative y-direction $\mathbf{f}\left(f_x, f_y\right) = (5, -10)$. The damping was incorporated into the MSM by expression (4.18) with scale factor $\alpha = 20$ resulting in $c^{ref}_{edge} = 17.88$ and $c^{ref}_{diag} = 28.84$ for the reference model and $c^{msm}_{edge} = 24.24$ and $c^{msm}_{diag} = 29.07$ for MSM derived. The Figure 4.3 shows the time evolution for errors computed by expressions (4.20) and (4.21) in sub-figure (a), and deformation behaviour in sub-figure (b). From these figures we conclude that the derived model has a very small error over time, given by millimeters in order of magnitude. Furthermore, the deformation behavior is practically identical over time for both the MSM models.

Experiment II) *Analyzing the influence of α and damping effect.* From the literature we know if the spring stiffness is assigned so that it varies inversely with the rest length, a reduction of the length directly implies a larger stiffness and, by consequence, a smaller time step (PALOC; FARACI; BELLO, 2006). Observing our expression (4.18) to calculate the damping constants c_{ij} we can apply this same reasoning to it. In previous experiment we had to use a time step $\triangle t = {}^{1}/_{480}$ linked to an $\alpha = 20$. Therefore, considering the relationship between spring stiffness and rest length, we conduct a experiment to clarify the influence of α in the time



(a)



(b)

Figure 4.3: Time integration analysis: (a) The percentage of root mean square error $\%e_{rms}$, and the percentage of maximum error $\%e_{max}$; (b) The deformation of MSM reference and the MSM derived

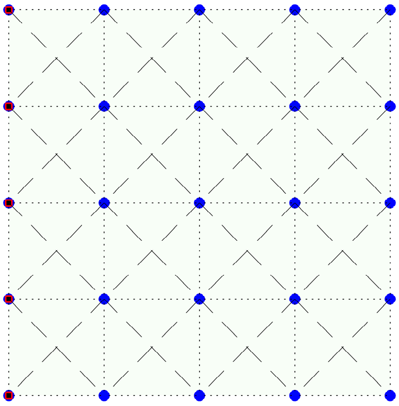


Figure 4.4: The MSM mesh for 5×5 elastic membrane. Fixed nodes are colored in red darkness and free nodes are colored as blue. The nodes are linked by stretch (dotted black) and shear (dashed black) springs.

step $\triangle t$. We instantiate a MSM model by setting up the masses $m_i = 2kg$, the stiffness $k_{edge} = 500$, $k_{diag} = 400$, $l^0_{edge} = 0.50m$, $l^0_{diag} = 0.7071m$, grid size of 2×2 meters, mesh resolution of 5×5 nodes and 72 springs as shown in Figure 4.4. We used numerical integration scheme with Leapfrog method and time step $\triangle t = 1/60$ initially.

In the first round of simulations we have kept constant the time step $\triangle t = 1/60$. So, we modify the α factor for $\alpha = 40, 60, 80, 100$ generating respective time integrations. Figure 4.5 shows deformation perceptual in time for theseα values. In this case, we noted numerical instability for $\alpha < 40$. In the second round of simulations, we set $\alpha = 1$ and test different values for $\triangle t$. Here, we experimentally noted that for $\alpha = 1$ we get numerical stability only when $\triangle t \leq 1/3000$.

From vibration theory (KELLY, 2011) we know that damping ratio $\zeta_i = 1$ assumed by Paloc (PALOC et al., 2002) is said to be critically damped, that mean that the system quickly reaches the static-equilibrium position. So, the larger damping and stiffness carry the system to stiff state, this in turn requires a tiny time step. Thus, the α role is to bring the system for a underdamped state. The underdamped state can be seen in Figure 4.5 where a damping graduate reduction is shown as α increases.

The experiments III to V validate our proposal when considering the linear elastic FEM as the reference model which is accurate only for small displacements (the global deformation in expression 4.17 is $\%\epsilon_G < 10\%$). In these experiments we make use of elastic objects in 2D and 3D. For 2D case we use an elastic quadrilateral and circular

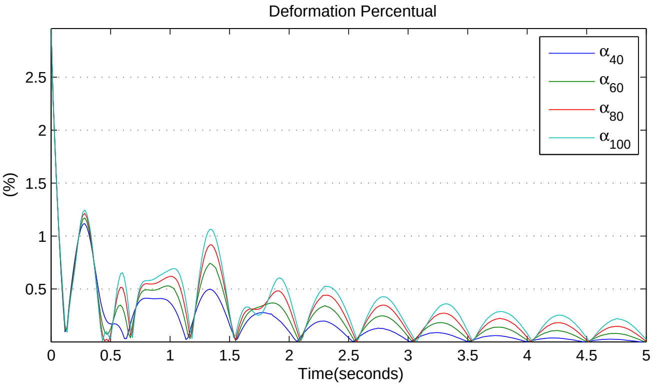


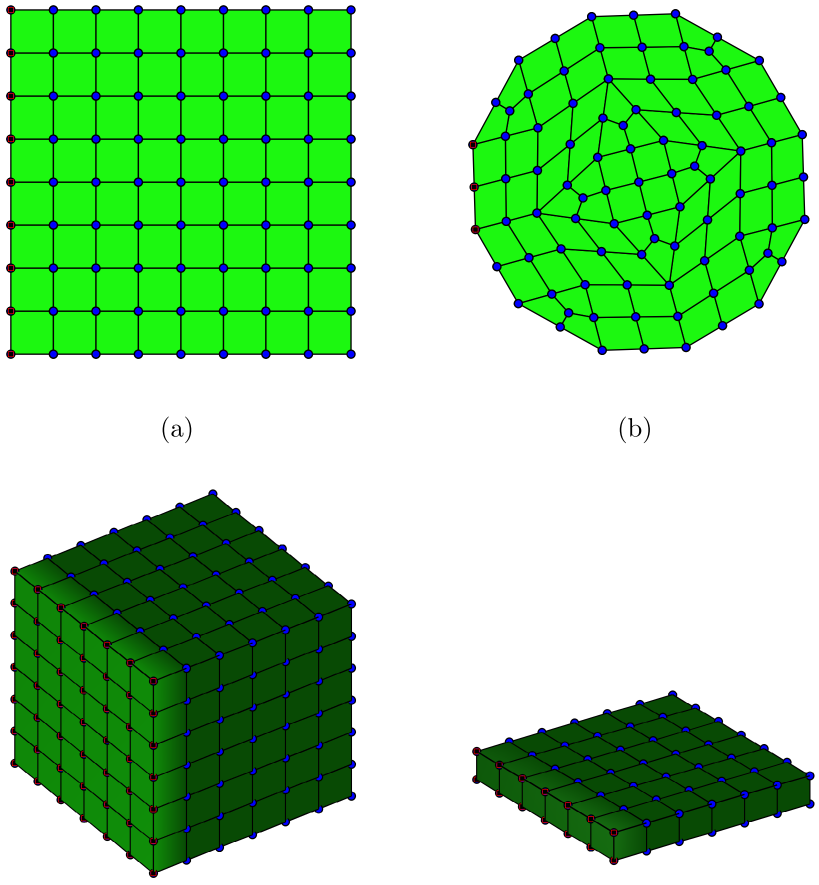
Figure 4.5: The time evolution of deformation perceptual for the MSM with $\triangle t = {}^{1}\!/_{60}$ and $\alpha = 40, 60, 80, 100$

membrane, and for 3D case we instantiate a cube and a plate. The size of the objects and the mesh resolution used for discretization have been shown in Table 4.1. Supplementing this information, Figure 4.6 shows the FEM meshes for each object.

For the experiments III to V, at the initialization of our method's procedures (step 1), we define material properties Young's modulus of 14.63 kPa and the Poisson's ratio of ${}^{1}\!/_{3}$, unless otherwise stated. We used numerical integration scheme with central difference method and time step of $\triangle t = {}^{1}\!/_{160}$ to simulate FEM equation (2.9). We follow an elasticity model for homogeneous, linear and isotropic material. We apply an isoparametric Q4 element in 2D case and B8 element in 3D case (the reader is referred to the references (COOK, 2001; ZIENKIEWICZ; TAYLOR; ZHU, 2005) for more about these elements).

Experiment III) *Performance comparative against the matrix approach.* . After the execution of our method's procedures we obtain the MSM parameters which are shown in Table 4.2.

In this experiment, we call "Matrix Approach" the methods present in (LLOYD; SZéKELY; HARDERS, 2007; NATSUPAKPONG; ÇAVUSOGLU, 2010; OTAMENDI, 2011) which uses expression 3.17 to compute spring constants. We must take into account that the matrix approach derives the parameters of MSM in the case of static equilibrium and, consequently, they do not consider damping effects (LLOYD; SZéKELY; HARDERS, 2007; NATSUPAKPONG; ÇAVUSOGLU, 2010; OTAMENDI, 2011). So, we have to solve the static equilibrium problem for FEM and MSM models to compare our method with the matrix approach. For this, a traction force is applied to the free nodes in positive x-direction and negative y-direction $\mathbf{f}\left(f_x, f_y\right) = (250, -400)$ for 2D case and $\mathbf{f}\left(f_x, f_y\right) = (1000, -1000)$ for 3D case, causing deformation below 10%. The Euclidean distance between FEM and MSM nodes in the corresponding equilibrium positions are



(a)

(b)

(c)

(d)

Figure 4.6: The FEM meshes for (a) square and (b) circular membranes, (c) cube and (d) plate. Fixed nodes are colored in red darkness and free nodes are colored as blue.

Objects	Spring	Coefficients		
Square Membrane	$k^{\bar{t}}_{edge}$	6228.075		
	k^{t}_{diag}	5086.067		
Cube	k^{t}_{edge}	500.898		
	k^{t}_{diag}	$554.145^{(a)}$	$929.238^{(b)}$	
Plate	k^{t}_{edge}	$156.564^{(a)}$	$3492.760^{(b)}$	
	k^{t}_{diag}	$948.079^{(c)}$	$1548.904^{(d)}$	$624.774^{(e)}$

Table 4.2: MSM parameters derived by acceleration approach. The $k^{\bar{t}}_{diag}$ in cube are face(a) and internal(b) diagonals. The $k^{\bar{t}}_{edge}$ in plate are stretch springs in (a) x-direction and y-direction and (b) z-direction while the $k^{\bar{t}}_{diag}$ (c) are face diagonals in xy-plane, (d) are face diagonals in xz-plane and yz-plane, and (e) are internal diagonals.

used for error evaluation. We report the error $\%e_{rms}$ and $\%e_{max}$, computed by expressions (4.20)-(4.21), respectively, in Table 4.3. The comparison with the error obtained with the matrix approach(NATSUPAKPONG; ÇAVUSOGLU, 2010; OTAMENDI, 2011) shows that the proposed technique performs better in these tests, except for the cube.

Approaches	Membrane		Cube		Plate	
	$\%e_{rms}$	$\%e_{max}$	$\%e_{rms}$	$\%e_{max}$	$\%e_{rms}$	$\%e_{max}$
Matrix Approach	8.19	15.82	11.91	22.70	27.03	64.47
Our	5.59	7.92	13.31	33.06	12.12	24.59

Table 4.3: Comparative results with matrix approach from references (LLOYD; SZéKELY; HARDERS, 2007; NATSUPAKPONG; ÇAVUSOGLU, 2010; OTAMENDI, 2011)

Experiment IV) *Effects of the geometry discretization.* Our purpose in conducting this experiment is to investigate how different discretization of mesh can affect the MSM parametrization. For this, we made a new discretization for the square membrane which geometry data is kept in Table 4.1. Figure 4.7 shows this new discretization. We take the FEM mesh of circular membrane already displayed in Figure 4.6 (b). So, we executed our method's procedures on these objects with time step of $\triangle t = {}^{1}/_{180}$ to simulate FEM equation (2.9).

In the case of quadrilateral membrane, we apply the expressions (4.11)-(4.13) with $\overline{m} = 6$, $l_1^0 = 0.125$, $l_2^0 = 0.165$, $l_3^0 = 0.253$, $l_4^0 = 0.350$, $l_5^0 = 0.460$, $l_6^0 = 0.555$. For $t = \bar{t}$, the MSM parameters obtained are: $k_1 = 2.297 \times 10^4$, , $k_2 = 5.968 \times 10^3$, $k_3 = 6.235 \times 10^3$, $k_4 = 1.034 \times 10^4$, $k_5 = 3.289 \times 10^3$, $k_6 = 6.417 \times 10^3$. For the circular membrane we

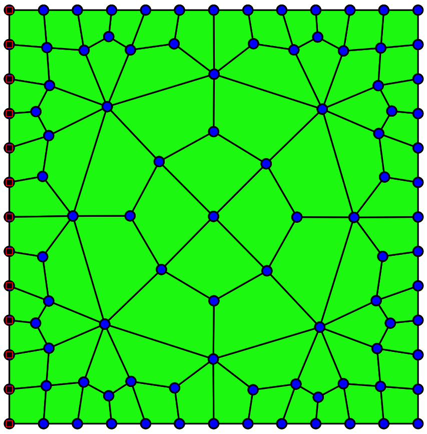


Figure 4.7: New discretization for square membrane with non-regular mesh. Nodes colored as red are fixed and those colored as blue are free.

ν	Stiffness		Error	
	k_{edge}	k_{diag}	$\%e_{rms}$	$\%e_{max}$
0.10	5308.765	6667.058	4.86	10.12
0.33	6228.075	5086.067	5.59	7.92
0.43	5658.309	5856.511	10.74	17.28

Table 4.4: Stiffness coefficients and errors for different Poisson's ratio.

cluster the springs into $\bar{m} = 24$ groups with $0.094 \leq l_i^0 \leq 0.445$, and the values of stiffness coefficients are $1.537 \times 10^3 \leq k_i \leq 2.107 \times 10^4$, $i = 1, 2, \ldots, 24$.

The validation of the results is similar to the previous experiments, ie, we solve the static equilibrium problem for FEM and MSM using the traction forces that act on free nodes. Both square and circular membrane were subjected to the traction force of $\mathbf{f}(f_x, f_y) = (250, -400)$. The obtained errors are: $\%e_{rms} = 2.42$ and $\%e_{max} = 8.77$ for square membrane; and $\%e_{rms} = 12.706$ and $\%e_{max} = 22.49$ for circular membrane.

By comparing the values for the quadrilateral membrane with the ones reported on Table 4.3 we can analyse the effect of the mesh type in the derived MSM. We notice that for the non-regular mesh the percentage of error falls in the range $[0, \%e_{rms} + \%e_{max}] = [0, 11.19]$ while for the regular one the corresponding interval is $[0, 13.51]$, which indicates a superiority of the former against the latter.

Experiment V) *Sensitivity to the Poisson's ratio value.* We conduct this experiment to investigate how Poisson's ratios can affect the MSM parametrization. We address this task by simulating the same object of Experiment I with Poisson's ratio $\nu \in \{0.10, 0.33, 0.43\}$. After the execution of our method's procedures the MSM parameters obtained were summarized in Table 4.4.

We carry out the validation of the obtained solutions following the same procedure used in previous experiments. We apply a traction force to the free nodes in positive x-direction and negative y-direction $\mathbf{f}\left(f_x, f_y\right) = (250, -400)$. and solve the static equilibrium problem for FEM and MSM models. The error $\%e_{rms}$ and $\%e_{max}$, computed by expressions (4.20)-(4.21), respectively, are also reported on Table 4.4. If compared with $\nu = 0.33 \approx 1/3$, we observe that $\%e_{rms}$ only decreases for ν=0.10 and $\%e_{max}$ always gets larger. Therefore, the error range increases for $\nu \neq 1/3$.

4.6 DISCUSSIONS

Carefully watching our experiments it is very clear that the great indicator of quality are the percentage of root mean square error $\%e_{rms}$, and the percentage of maximum error $\%e_{max}$. These two measures quantify the difference between the reference model and the simulated model in percentage terms. A high value means big difference between the two models, that is, low similarity of behavior. So, our discussion is constantly referencing the values for these measures.

In Experiment I, we were interested in checking our method in the MSM parameterization using a nonlinear model as reference. In this case, we choose the MSM itself as reference model with square elastic membrane due to ease of set up the physical parameters of model. The results, despite the mesh simplicity are very convincing about of the efficiency of the acceleration based approach. The observed values $\%e_{rms} = 0.69$, $\%e_{max} = 1.20$ attest that exists very little difference between the two models. That facto can also be seen in the comparison of the time evolution between the two models.

While performing temporal integration of the first experiment, we note different time steps between the two temporal integrations. The time step for the temporal evolution of parameterization was of $\triangle t = 1/120$ and the time step for the temporal evolution of experiment was of $\triangle t = 1/480$. The temporal evolution of parameterization has not damping thus we deduce that the damping effect could bring numerical instability for the temporal integration. This was the motivation for the Experiment II.

The experiment II was in fact a justificative for adding α into the expression of the damping computation which appears in the reference (PALOC; FARACI; BELLO, 2006). From arguments already exposed in the experiment itself, we demonstrate that the term α controls the damping effect and indirectly ensures numerical stability during temporal integration.

Following a common practice in the literature (LLOYD; SZéKELY; HARDERS, 2007; BAUDET et al., 2009; NATSUPAKPONG; ÇAVUSOGLU, 2010; OTAMENDI, 2011; SILVA; GIRALDI; APOLINARIO, 2015), in experiments III to V we parameterize the MSM using the linear elastic FEM as reference model. That means that the reference model follows the constitutive equations (see section 2.1) based on elasticity theory for small deformations. Compared to other methods of literature (Experiment III) the acceleration based approach achieved better results.

The experiment IV has an interesting aspect to consider that is how different discretizations affect the quality of the MSM parameterization. It is important to comprise that the discretizations of this experiment are more complex than previous ones. In these discretizations the rest length of springs are heterogeneous. Faced this diversity in rest length, we defined a heuristic to cluster the springs by its rest length (as detailed in section 4.2). The results demonstrate that the combination of the heuristic with a mesh refinement (differente discretization) improved the results.

As one may notice in experiments I-II, we have used the MSM as reference model. On the other hand, experiments III-V were conducted by using the linear elastic FEM as reference model. Therefore, it is demonstrated that the aceleration based approach is not restricted to a specific model.

A limitation that was presented by the approach described in this chapter was the difficulty to generate acceptable results with complex geometries in 3D case. In the next chapter, we will treat of this issue in the context of isogeometric analysis which provides well adapted tools to handling of complex geometry.

Chapter 5

Non-uniform Rational B-Spline (NURBS) is an well-established geometric tool with industry standard for representing curves, surfaces and solids quite accurately. The Isogeometric Analysis (IGA) is a FEM-like method that uses NURBS not only for the description of the geometries but also for the solution of field variables. The IGA allows using of higher order elements and thereby opens up possibilities to designing the MSM parameters for the applications that have interactive mechanical simulations such that the geometry and derived parameters inherit the accuracy from NURBS-based IGA. In this chapter we propose a NURBS-based MSM wherein control points are treated like mass points which are connected by massless springs. In this proposal the stiffness coefficients are computed by using data-driven strategy with a NURBS-based IGA model acting as reference model.

PARAMETERIZATION BASED ON IGA

The concern with accuracy and realism is a central theme in the simulation of deformable objects. For models based on continuous like FEM, an improvement of accuracy and consequently of realism can be achieved with mesh refinements in which the discretization of the region can be improved by r-method, h-method or p-method (RAO, 2004; ZIENKIEWICZ; TAYLOR; ZHU, 2005). A discrete model such as MSM has an inborn inaccuracy because in general is not directly based on continuum mechanics. However, it is possible to apply mesh refinement based on the techniques r-method and h-method to improve accuracy of simulation. On the other hand, due to its discrete nature a approach that uses the p-method concept cannot be applied.

To deal with complex geometries and problems involving curved boundaries the FEM-based techniques can use higher order elements and isoparametric concept. These tools can improve accuracy to both geometry description and the solution field variable. The discrete models do not have a similar feature to improve accuracy.

To overcome the limitations of the MSM the researchers have been trying to incorporate the accuracy and realism from continuous models into the MSM by the parametrization process. However, the MSM parameterization from a continuous model with higher order elements has not been explored, at least until the date of the last literature reviewing. The contact with the isoparametric FEM and IGA led us to research the development of a method that could fulfill this gap.

As seen in Chapter 2, the IGA framework allows to use NURBS to represent both geometry and physical fields in the solution of problems governed by partial differential equations (PDE) and also enable higher-order polynomial basis functions which should make simple the using of the FEM with higher order elements. It integrates the computer aided design tools with finite element analysis into the same unified framework and provides a very robust approach for numerical modeling with highly precise geometry representation by using B-Splines and NURBS parametrization (COTTRELL; HUGHES; BAZILEVS, 2009). These features can be exploited to improve the quality of derivation process, the accuracy of geometry and the modeling of the deformable objects for the MSM.

The characteristics of the IGA motivated us to propose a mass-spring model NURBS-based wherein control points are treated like mass points which are connected by massless springs. Hence, we term it of NURBS Mass-Spring Model, abbreviated to NMSM. To setting NMSM parameters (mass, spring stiffness coefficient and damping constants) we proposed a new data-driven strategy with a NURBS-based IGA model acting as continuum counterpart model. Our method computes the spring coefficients solving an optimization problem based on the static equilibrium of NURBS-based IGA model and NMSM; calculates particles mass following (BAUDET, 2006; LLOYD et al., 2008) and derives the damping constant using a heuristic based on (PALOC et al., 2002). With this approach, we increase the geometric accuracy in the representation of elastic objects, since we have a mass-spring model based on NURBS; we can simulate deformation especially of curved shapes by exploring the NURBS features to refinement or manipulation of high-order polynomial which improves smoothness in the deformations; we combine the simplicity and computational efficiency of the MSM with NURBS to achieve deformation simulation in real-time with realism similarly to the offered by NURBS-based IGA model ; and finally we determine the NMSM parameters so that it behaves like IGA (its counterpart continuum) eliminating the trial-and-error approach for the parameters tuning.

5.1 THE NURBS MASS SPRING MODEL

Up to our actual knowledge of the MSM literature, a proposal that combines IGA, NURBS and MSM is innovative because it uses the IGA model in the MSM parametrization exploring the IGA qualities such that: geometric accuracy, simplicity to modification of the polynomial order in the basis functions, mesh refinement, relationship with constitutive laws of material, and mainly be able to perform the MSM parametrization applying a FEM with higher order elements. Therefore, the formulation that we will make in the next sections is an important contribution of this thesis.

Definition 17. Be $\mathcal{M} = (V, E)$ a graph to the generic particle mesh where $V = \{v_1, \ldots, v_n\}$ is the set of vertices and $E = \{e_1, \ldots, e_m\}$ is the set of edges, formed by pairs of vertices, such that

$$E\left(V\right) = \left\{\left(u, v\right) \mid u, v \in V, u \neq v\right\} \text{ with } E \subseteq E\left(V\right).$$

We define the MSM mesh as an edge-weighted graph formed by the pair $(\mathcal{M}, l^0)$ where $l^0(e) : E \rightarrow \mathbb{R}+$ is the rest length associated with each edge in E, each vertex $v_i \in V$ is linked to a mass m_i and position $\mathbf{x}_i$, such that $v_i \rightarrow (m_i, \mathbf{x}_i)$, and the edge $e_j \in E$ maps the interconnection springs and dampers by the incidence matrix $B = (b_{ij})_{n\times m}$ defined on the graph $\mathcal{M} = (V, E)$, given by

$$b_{ij} = \begin{cases} l^0_{ij} & if\ v_i \in e_j \\ 0 & otherwise \end{cases}, \tag{5.1}$$

for which the Equations (3.5) and (3.6) hold true.

In our NMSM, the MSM mesh is built upon in the NURBS formulation where the control points are turned into masses points connected by the massless springs network with natural length greater than zero. Formally, we define the NMSM mesh as follow:

Definition 18. Let the generalized geometry mapping that according to the Definition 12 is given

$$\mathbf{g}(\boldsymbol{\xi}) = \sum_{\mathbf{i}\in\mathcal{I}_b} R_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi})\,\mathbf{p}_\mathbf{i}, \tag{5.2}$$

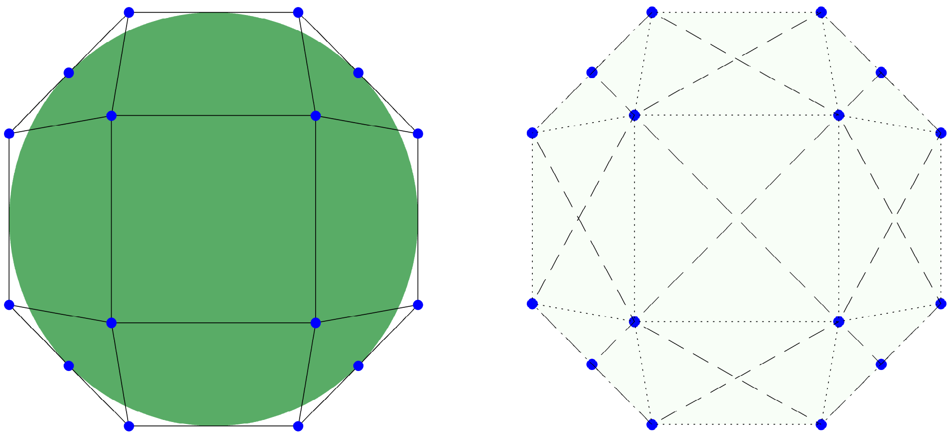
where $\mathbf{i} \in \mathcal{I}_b$ are d-dimensional multi-indexes, $\boldsymbol{\xi} = \left(\xi^{(1)}, \ldots, \xi^{(d)}\right)$ and $\mathbf{p} = (p_1, \ldots, p_d)$ are the coordinate values and the polynomial order along each parametric direction, and $\mathbf{p}_\mathbf{i} \in \mathbb{R}^3$ is the control mesh. We define the NMSM mesh as an edge-weighted graph $(\mathcal{M}, l^0)$ where $v_\mathbf{i} \rightarrow (m_\mathbf{i}, \mathbf{p}_\mathbf{i})$ with $\mathbf{i} \in \mathcal{I}_b$. To make a counterpoint to the control mesh $\mathbf{p}_\mathbf{i} \in \mathbb{R}^3$, we termed the NMSM mesh as the spring control mesh.

For NMSM meshing from NURBS based IGA model we use the mapping similarly to that shown in Figure 4.1. But this time, the FEM nodes are turned into control points. Thus, in Figure 4.1, the nodes on left side in figure form the control mesh which is mapped to spring control mesh (right side in figure) which in addition to control points contains stretch (edge) and shear (diagonal) springs.

The geometry complexity can degenerate quadrangle or hexaedron in the mapping of Figure 4.1. For instance, in Figure 5.1(a) we have shown a circular membrane of unit radius, with 16 control points. The associated shape functions are 2D NURBS, given by expression (2.25), with polynomial order $p = q = 2$. The mapping pictured in Figure 4.1 generates the NMSM mesh shown on Figure 5.1 (b).

Analogously to linear IGA under small deformation scenario, we can derive the linearized NMSM equations of motion. In order to perform this task we must linearize the NMSM governing equation by using the first order approximation for the force of the spring that connects nodes i and j around the rest nodal position $\left(\mathbf{p}_i^0, \mathbf{p}_j^0\right)$ that results in the expression

$$\mathbf{f}(\mathbf{p}_i, \mathbf{p}_j) \approx \mathbf{f}_{(i,j)} + \left[\frac{\partial \mathbf{f}(\mathbf{p}_i, \mathbf{p}_j)}{\partial \mathbf{p}_i}; \frac{\partial \mathbf{f}(\mathbf{p}_i, \mathbf{p}_j)}{\partial \mathbf{p}_j}\right] \left[\begin{array}{c} (\mathbf{p}_i - \mathbf{p}_i^0) \\ (\mathbf{p}_j - \mathbf{p}_j^0) \end{array}\right]. \tag{5.3}$$



(a) (b)

Figure 5.1: Mapping IGA to NMSM mesh. The control mesh (pictured in blue and black line) and its corresponding physical mesh (fiiled in green) in sub-figure (a) is mapped to spring control mesh (b). In both meshes the control points are pictured in blue points. The stretch (dotted black) and shear (dashed black) springs are shown in (b).

The first term of expression (5.3) can be discarded because $\mathbf{f}_{(i,j)} = \mathbf{f}\left(\mathbf{p}_i^0, \mathbf{p}_j^0\right) = -\mathbf{f}\left(\mathbf{p}_j^0, \mathbf{p}_i^0\right)$, and so, they will cancel each other when computing the resultant force. The process of assembling the linearized equations for the springs gives a symmetric stiffness matrix K_{NMSM}, and, consequently, the linearized NMSM that has the governing equation

$$M_{NMSM}\ddot{\mathbf{x}} + K_{NMSM}\mathbf{x} = \mathbf{f}_{ext}, \tag{5.4}$$

where $\mathbf{x} = \widehat{\mathbf{p}} - \widehat{\mathbf{p}}_0$, the mass matrix $M_{NMSM} = diag\left[\begin{array}{cccc} m_1 I & m_2 I & \ldots & m_{n_p} I \end{array}\right]$, $\widehat{\mathbf{p}}$ is an $n \cdot d \times 1$ vector that contains all the d-dimensional control points positions $\mathbf{p}_i$, and $\mathbf{f}_{ext}$ is the vector that holds the action of external forces over the NMSM modes.

Similarly to Equation (3.5), the Equation (5.4) need initial conditions composed by the initial configuration and velocity to assure existence and uniqueness of the solution. As can seen in the section 3.1, these conditions will be the input to the Leapfrog method for time integration.

5.2 NMSM PARAMETERIZATION BY IGA

In Figure 5.2, we show an overview of the main elements of our proposal. In general way, our methodology uses the IGA-based model with all its theoretical apparatus. For this reason, we dispose of NURBS framework for the object geometry representation. On left side of Figure 5.2, we have control mesh, the underlying geometry (physical mesh) and the material properties. On right side of Figure 5.2, it can be seen the spring control mesh and its parameters (mass, spring stiffness coefficient and damping constants). It should be noted that the underlying geometry (physical mesh) of the IGA model is identical to

NMSM. The geometric match between the NMSM and the NURBS based IGA framework is performed following the mapping in Figure 4.1, as represented by the blue arrow in Figure 5.2. Hence, NURBS control points are mapped into particles masses with the NMSM topology defined according to Figure 4.1.

From the view point of processing, Figure 5.2 depicts a sequence of steps that starts with the construction of the geometry NURBS which depends on knot vectors, NURBS basis functions and their respective polynomial orders; and continue with the definition of the material properties which concludes the IGA model configuration (left side of figure). Next, whole NURBS apparatus is transferred to the NMSM that uses the NURBS data to generate the spring control mesh (represented by the blue arrow in Figure 5.2). The last stage is the NMSM parametrization itself. Initially, a minimization problem involving the function $\phi = \min\left(f_{IGA} - f_{NMSM}\right)$ is solved to compute stiffness coefficients, where the functions f_{IGA} and f_{NMSM} encode characteristics of the two models. The mass density ρ from IGA model is used to compute particles masses, and finalizing stiffness coefficients and masses are combined to yield damping constants.

In FEM analysis of linear elasticity, the properties of an elastic material are defined from Young's modulus and Poisson's ratios (ZIENKIEWICZ; TAYLOR; ZHU, 2005). As illustrated in Figure 5.2, we expect to map these properties to spring coefficients. For this, we shall formulate a minimization problem that involves an objective function that depends on the difference between functions f_{IGA} and f_{NMSM} which are used to measure the similarity between the two models. Besides, the proposed method computes the nodal masses by using a function f that depends on mass density ρ, and the damping constants are calculated by an expression that includes the mass and spring coefficients. In summary, our goal is to parametrize the NMSM model such that it behaves as NURBS based IGA model.

In our approach to compute spring coefficients, we follow the static equilibrium reasoning (see Equations (3.15)- (3.17), section 3.2). However, we develop a new objective function that measures the similarity between the reference model and the linearized NMSM displacement by using their stiffness matrices. In static equilibrium we shall observe that the linearized NMSM equation and the linearized IGA, calculated by expressions (5.4) and (2.9), respectively, simplifies to :

$$\mathbf{K}_{NMSM}\widehat{\mathbf{u}} = \mathbf{f}_{ext} \Longrightarrow \widehat{\mathbf{u}}^{nmsm} = \left(\mathbf{K}_{NMSM}\right)^{-1}\mathbf{f}_{ext}, \tag{5.5}$$

$$K_{IGA}\widehat{\mathbf{u}} = \mathbf{f}_{ext} \Longrightarrow \widehat{\mathbf{u}}^{ref} = \left(\mathbf{K}_{IGA}\right)^{-1}\mathbf{f}_{ext}, \tag{5.6}$$

where $\widehat{\mathbf{u}} = \widehat{\mathbf{p}} - \widehat{\mathbf{p}}^0$. So, if we build the vector $\boldsymbol{\phi} = (k_1, k_2, \ldots, k_{\bar{m}})^T$that contains the stiffness coefficients then $\mathbf{K}_{NMSM}$ becomes a functional matrix of $\boldsymbol{\phi}$. Consequently, we can postulate that the optimum $\bar{\boldsymbol{\phi}}$ is given by the solution of the optimization problem

$$\bar{\boldsymbol{\phi}} = \arg\min_{\phi}\left[\left(\frac{\sqrt{\frac{1}{n_p}\sum_{i=1}^{n_p}\left\|\mathbf{u}_i^{ref} - \mathbf{u}_i^{nmsm}\right\|^2}}{\max_i\left(\left\|\mathbf{u}_i^{ref}\right\|\right)}\right)\cdot 100\right], \tag{5.7}$$

where $\mathbf{u}_i^{ref} = \mathbf{p}_i^{ref} - \mathbf{p}_i^0$, $\mathbf{u}_i^{nmsm} = \mathbf{p}_i^{nmsm} - \mathbf{p}_i^0$, are computed by equations (5.5)-(5.6), and n_p is the number of control points, as explained in Chapter 2.

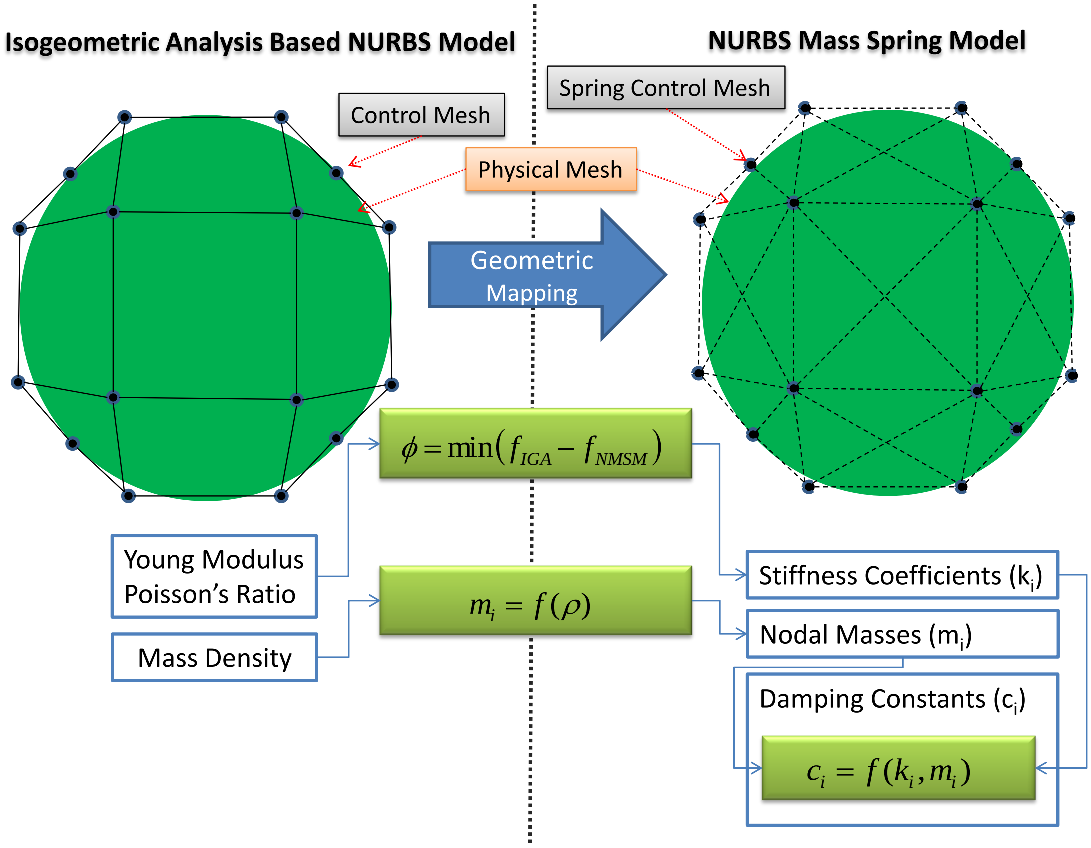


Figure 5.2: Overview of mapping from the NURBS-based IGA model to NMSM. On the left side is gathered control mesh, material properties and NURBS apparatus. On the right side is gathered spring control mesh with stretch and shear springs and its parameters computed by means specific functions. The underlying geometry (physical mesh pictured in green) is identical to both models.

Algorithm 5.1 The NMSM Parametrization

```
InitModels();
Gamma := MaxGamma;
BestGroup := GroupByRest(Gamma);
Error := OptimizePhi(BestGroup);
While Gamma>GammaStep do
    newGroup := GroupByRest(Gamma);
    newError := OptimizePhi(newGroup);
    If newError < Error Then
        Error := newError;
        bestGroup := newGroup;
    End
    Gamma := Gamma − GammaStep;
End
ComputeMasses();
ComputeDamping();
```

Analogous to acceleration based approach (section 4.1), we compute the masses m_i and damping constants c_{ij} using expression (3.12) and (4.18). The Algorithm 5.1, summarizes the main steps of the proposed method to NMSM parametrization.

The Algorithm 5.1 starts running procedure "initModels" which sets up the material properties (the physical characteristics of deformable objects which comprises the Young modulus, Poisson's ratios and mass density) into the IGA model and keeps data required for the NURBS setting as control points, polynomial order and knot vectors to both models (IGA and NMSM). We have observed that the NMSM is set with the same NURBS setting assigned to IGA model. At this point it is necessary a geometric mapping to create the springs that will connect the NMSM mesh points. Figures 4.1 and 5.2 show the mapping between IGA geometry and NMSM mesh.

After defining geometry and material properties, the algorithm enters a loop to find out the spring groups that make the two models behave similarly (eg, which group setting leads to the lower error between the two models). This task begins with the procedure "groupByRestLength" that will perform the springs clustering process, using the granularity factor γ and following equations (4.11)-(4.13). In the sequel of the grouping procedure it takes place solving the optimization problem of expression (5.7) to calculate the spring coefficients. In the first iteration we assign a maximum value to factor γ leading to the smaller group size. So, we iteratively solving problem in expression (5.7) and decreasing γ until you find the group setting with minimum error.

The expression (4.19) in section 4.4 is an upperbound for each spring i, j. On the other hand, after to apply the grouping process in all springs, it is necessary to select

Object	Size(meter)	$(n \times m \times l)$	(p, q, r)	Springs
Square membrane	2×2	6×6	$(2, 2)$	110
Circular membrane	radius=1	6×6	$(2, 2)$	110
Elliptical membrane	major radius=1, minor radius=0.5	6×6	$(2, 2)$	110
Elastic cube	$2 \times 2 \times 2$	$6 \times 6 \times 6$	$(2, 2, 2)$	1940
Elastic torus	inner radius=0.5, external radius =1.25	$9 \times 5 \times 5$	$(2, 2, 2)$	2000
Coupled spheroid	length=4, radius=0.5	$9 \times 10 \times 3$	$(2, 2, 1)$	2315

Table 5.1: The geometry data of elastic objects used in experiments. Geometry column contains data abou size(meters), control mesh $(n \times m \times l)$, NURBS polynomial order (p, q, r), and number of springs.

a value to represent the group. We have defined that the upperbound for the group is given by

$$k_b^g = \max\left(\mathcal{G}_\alpha\right), \; \mathcal{G}_\alpha = \left\{k_b^{ij} \in \left\{f_\alpha\left(k_{i,s}\right) k_b^{ij}\right\}\right\}, \tag{5.8}$$

where $\mathcal{G}_\alpha$ is a set whose members are k_b^{ij} following the grouping rule in expression (4.13), that is, $k_b^{ij} \in \mathcal{G}_\alpha$ is the coefficient of the spring s that satisfies the relationship $l_{i,s}^0 \in [l_\alpha^0(1-\gamma), l_\alpha^0(1+\gamma)]$.

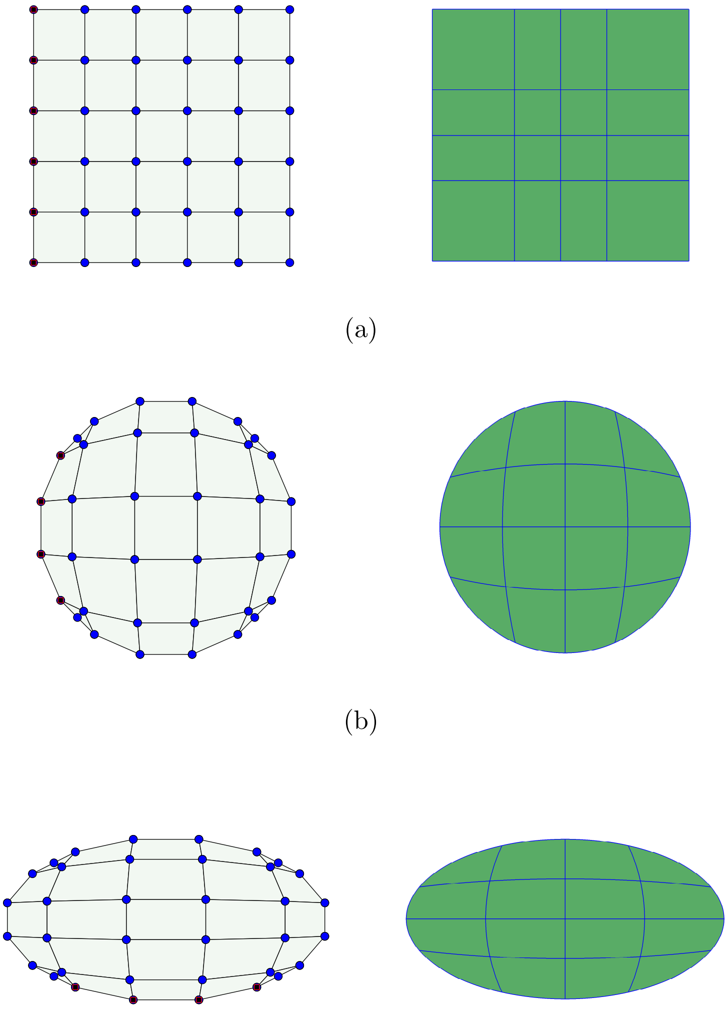
Finally, with stiffness constants computed, The Algorithm 5.1 calculates nodal masses by equation (3.12) and damping factor by using expression (4.18).

5.3 RESULTS

In order to validate the proposed of the NMSM parameterization, we conduct some experiments using 2D and 3D elastic objects. Initially, we aim to test how close the NMSM can behave of its continuum counterpart, the IGA model. Next, we verify the response of our algorithm in very specific situations: response to various mesh resolutions, behavior under variation of Poisson's ratios, sensitivity to the heuristic of grouping springs, and comparative results with other techniques of parametrization. Table 5.1 gathers the elastic objects and their corresponding geometries which were used in each experiment. The Figures 5.3 and 5.4 show the corresponding control points and physical meshes. The experiment conducted are: I) Similarity analysis with IGA and NMSM; II) Responsiveness to Mesh Resolution; III) Responsiveness to Grouping Spring; and IV) Comparing to Matrix Approach; and V) Damping and Dynamic Evolution.

The elastic objects used in our experiments have the Young's modulus given by $E = 15kPa$ which is an usual value for soft body (LLOYD; SZéKELY; HARDERS, 2007; NATSUPAKPONG; ÇAVUSOGLU, 2010), the Poisson's ratio of $\nu = 1/3$, and mass density $\rho = 1.143kg/m^3$ unless otherwise stated. In all the experiments, we have the following boundary conditions: control points colored as red darkness are fixed and a traction force resulting into a deformation of around 10 percent is applied to the free control points colored as blue (see expression (4.17) in Appendix **??** for more details about the computation of global deformation).

Experiment I) *Similarity analysis with IGA and NMSM.* In this experiment, we per-

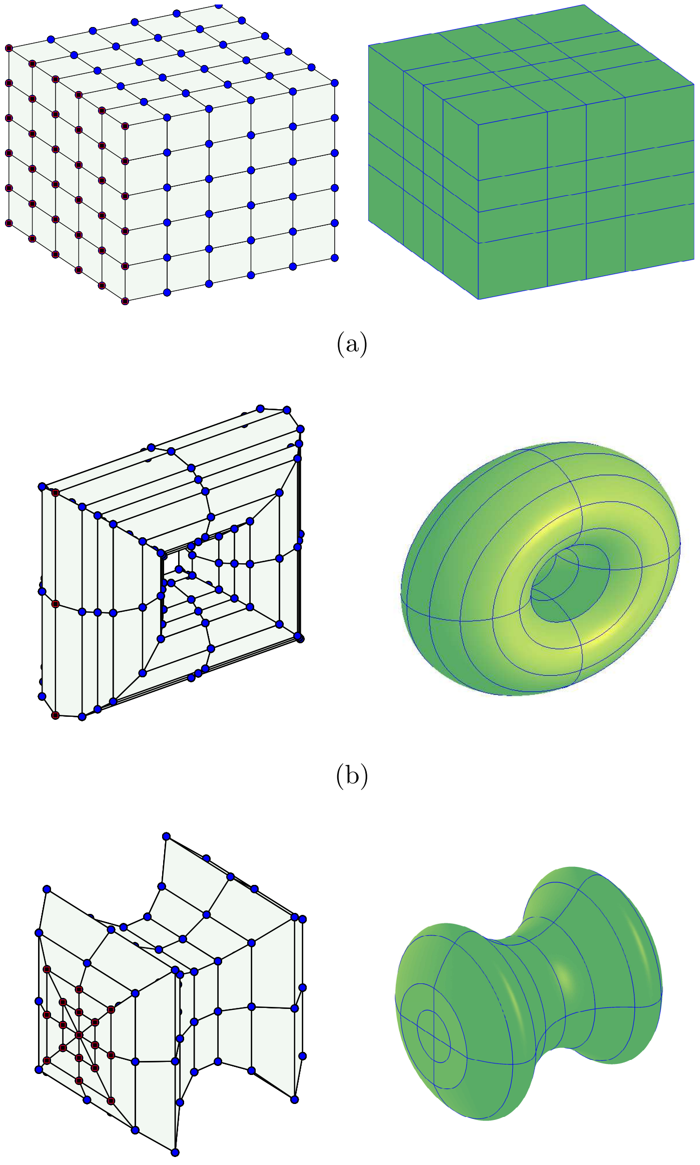


(a)

(b)

(c)

Figure 5.3: Control (on left) and physical (on right) meshes for 2D NURBS-based IGA elastic objects. a) Square; b) Circle; c) Ellipse. Fixed control points are colored in red darkness. Free control points are colored as blue



(a)

(b)

(c)

Figure 5.4: Control (on left) and physical(on right) meshes for 3D NURBS-based IGA elastic objects. a) Cube; b) Torus; c) Coupled Spheroid. Fixed control points are colored in red darkness. Free control points are colored as blue.

Object	$\mathbf{f}_{ext} = (f_x, f_y, f_z)$	Group Spring ($\bar{m}$)	Coefficients ($k_1 \dots k_{\bar{m}}$)
Square membrane	$(200, -200, 0)$	2	$(5360.352, 8624.863)$
Circular membrane	$(600, 100, 0)$	16	$(347.737 \dots 23116.788)$
Elliptical membrane	$(1600, 200, 0)$	28	$(276.385 \dots 47256.123)$
Cube	$(80, -40, 40)$	3	$(624.417, \dots, 3825.382)$
Torus	$(60, -30, 30)$	4	$(25.095, \dots, 52900.287)$
Coupled Spheroid	$(100, -50, 50)$	3	$(583.968, \dots, 47793.992)$

Table 5.2: Number of groups and spring coefficients obtained by Algorithm 5.1.

form tests with several elastic objects (2D and 3D cases) to verify the similarity of deformation between NMSM and IGA models. The tested objects and their geometries are indicated in Table 5.1 .

To compute NMSM parameters we have to solve the static equilibrium problem involving the IGA and NMSM models in order to find the response of the optimization problem in the expression (5.7). In this case, it is necessary to set the force vector in expression (4.9) . We have defining it to ensure deformation around of 10 percent in order to avoid degeneration of IGA elements. Table 5.2 shows the external force which is a traction force applied to the free nodes for each object together with the groups of spring and stiffness coefficients computed by Algorithm 5.1. One must note that we put the computed stiffness coefficients into ascending order.

Our results are analyzed quantitatively by computing the percentage of root mean square error $\%e_{rms}$, and the percentage of maximum error $\%e_{max}$ of Euclidean distance between reference control points and NMSM nodes (equations (4.20)-(4.21)). In Table 5.3 we report the error $\%e_{rms}$ and $\%e_{max}$ under applied force for the 2D and 3D cases, respectively. For 2D case, we can observe an error increase in circular and elliptical membrane while for 3D case the error increases in the torus and icoupled spheroid. An initial explanation for this behavior would be that this increase of error is a consequence of increase in the geometry complexity. However, other aspects need to be considered. We will discuss in due course.

Object	$\%e_{rms}$	$\%e_{max}$
Square membrane	1.14	2.79
Circular membrane	2.50	8.87
Elliptical membrane	2.44	6.67
Cube	1.73	7.44
Torus	14.28	31.37
Coupled Spheroid	5.37	22.01

Table 5.3: Quantitative results 2D/3D objects

Experiment II) *Sensitivity to Mesh Resolution.* To perform this experiment we choose the square membrane and elastic cube due to the simplicity of geometry. Using

	Resolution	$\%e_{rms}$	$\%e_{max}$
Square Membrane	3×3	14.37	22.69
	4×4	2.35	5.25
	6×6	1.14	2.79
	9×9	0.50	1.41
Elastic Cube	$3 \times 3 \times 3$	14.03	27.27
	$4 \times 4 \times 4$	2.94	6.25
	$6 \times 6 \times 6$	1.73	7.44
	$9 \times 9 \times 9$	1.70	4.60

Table 5.4: Sensitivity to mesh resolution mesh

the same setting of material and geometry previously shown, we modify the mesh resolutions to test the error behavior.

So, we just have modified the control mesh resolution, according to Table 5.4. We run the Algorithm 5.1 and compute the error $\%e_{rms}$ and $\%e_{max}$ which are reported in Table 5.4. From the NURBS viewpoint, the consequence of control mesh refinement is the splitting of existing patches into new ones without changing the continuity of the representation. This enhances the flexibility of the IGA model which leads us to expect that the error decreases as the mesh resolution increases, as noticed in most of the cases in Table 5.4.

Experiment III) *Sensitivity to Grouping Spring.* The experiment I) indicated that there is a relationship between complex geometries and the error rate. It is known that the mesh topology, the stiffness coefficients and rest lengths have a paramount role in the control the degrees of freedom of the system. On the other hand, a complex geometry implies in the increasing of the degrees of freedom which in turn may result in a heterogeneity of rest lengths with the corresponding increase of spring coefficients. So, one notes a clear relationship between the complex geometry and the parameters of the springs (rest length and stiffness coefficient). Therefore, a mechanism to manage these parameters can be an important tool in designing of the NMSM. Our heuristic to cluster the springs described in section 4.2 is this tool. In that section, we define the factor γ to deal with the rest length heterogeneity by controlling the number of springs groups. The current experiment is important to verify how changing in this parameter affect the results of the Algorithm 5.1 and can help in finding a tradeoff between geometry complexity and error rate.

To perform the test we consider that the object geometry must have some heterogeneity in its rest length, so we choose the elastic membranes with circular and elliptical shapes, and torus and coupled spheroid. With the default setting described above, we run the Algorithm 5.1 a number of times, once for each distinct value of γ. So, we derive the NMSM and compute the errors which are reported in Table 5.5.

An initial examination in Table 5.5 of the error rate for circular and elliptical membranes shows an enhancement of the solution when the group size (the number of springs

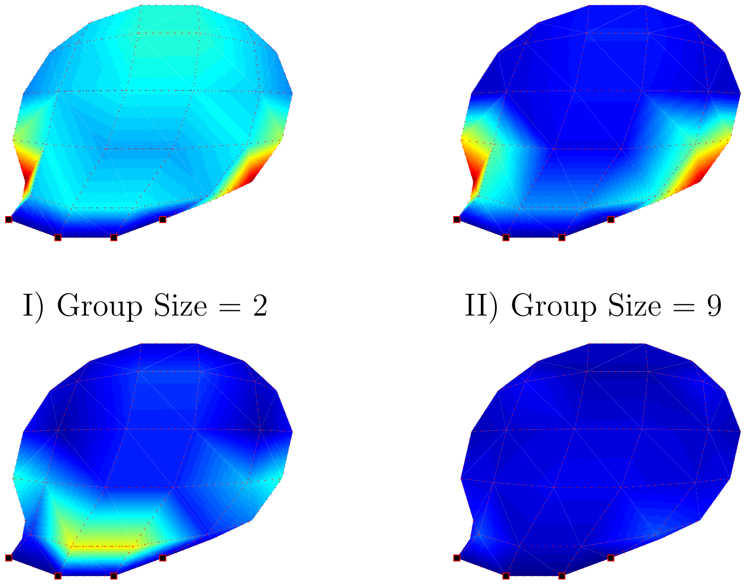
Object	Running	γ	Group Size ($\bar{m}$)	$\%e_{rms}$	$\%e_{max}$
Circular membrane	I)	0.800	2	5.56	25.20
	II)	0.200	6	5.14	19.44
	III)	0.010	16	2.50	8.87
	IV)	0.001	17	3.31	14.30
Elliptical membrane	I)	0.700	2	10.86	36.89
	II)	0.100	9	5.47	23.17
	III)	0.050	16	5.03	11.65
	IV)	0.001	28	2.44	6.67
Torus	I)	0.700	2	22.95	51.27
	II)	0.500	4	14.28	31.37
	III)	0.200	8	34.90	63.51
	IV)	0.100	16	43.80	75.30
Coupled Spheroid	I)	0.800	2	47.19	75.35
	II)	0.600	3	5.37	22.01
	III)	0.200	7	10.04	19.17
	IV)	0.010	60	34.62	61.56

Table 5.5: Grouping spring for the 2D geometry experiments

groups) becomes higher, except for running IV. On the other hand, there is no apparent relationship between group size and error for the runnings with torus and coupled spheroid objects. In this sense, we must not forget that in the MSM, the mechanical properties are consequences of the mesh topology, the stiffness coefficients and rest lengths, which control the degrees of freedom of the system. Springs being grouped by their rest length implies into a restriction that directly affects the mechanical behavior of system through the stiffness coefficient because as shown in expression (5.8) it is assigned a single value of stiffness to all the springs in the group. To escape these traps, our algorithm finds out the group setting that has minimum error.

We made an illustration that helps to see more clearly the group with the lowest error for the elliptical membrane and coupled spheroid objects. The Figures 5.5 and 5.6 and picture the difference distribution between NURBS based IGA and NMSM displacements rendered using colors that are scaled through the intensity of the difference $\mathbf{u}^{ref} - \mathbf{u}^{nmsm}$, where $\mathbf{u}^{ref}$ and $\mathbf{u}^{nmsm}$ are the static equilibrium solution given by expressions (5.6) and (4.9), respectively.

Experiment IV) *Comparing to Matrix Approach.* In the subsection 4.5 that is the results subsection of the acceleration based approach, we call "Matrix Approach" the methods present in (LLOYD; SZéKELY; HARDERS, 2007; NATSUPAKPONG; ÇAVUSOGLU, 2010; OTAMENDI, 2011). which uses expression (3.17) to compute



I) Group Size = 2

II) Group Size = 9

III) Group Size = 16

IV) Group Size = 28

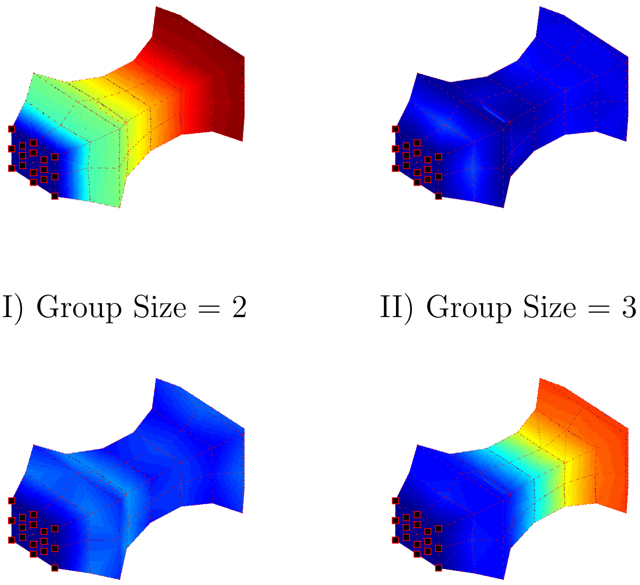
Figure 5.5: Error representation for elliptical membrane. Error color scale is 0 (blue) to 1.143×10^{-1} (red) meters

Object	NMSM		Matrix Approach	
	$\%e_{rms}$	$\%e_{max}$	$\%e_{rms}$	$\%e_{max}$
Square	1.14	2.79	47.57	68.82
Ellipse	4.49	12.35	39.00	80.74
Cube	1.73	7.44	93.25	190.79
Torus	14.28	31.37	337.87	706.02

Table 5.6: Comparative quantitative results

spring constants. One should not forget that the expression (3.17) measures the similarity between the NURBS based IGA model and the linearized NMSM stiffness matrices. For this comparison, we selected square and circular elastic membrane in 2D case, and cube and torus in 3D. Using the same setting of material and geometry previously shown in Table 5.1, the stiffness coefficients are computed solving the expression (3.17) and running the Algorithm 5.1. Table 5.6 reports the errors $\%e_{rms}$ and $\%e_{max}$ for the NMSM and the Matrix Approach. The error comparison shows that the proposed technique performs better in all tests.

From Table 5.6 we can notice that the error for Matrix Approach is very high. Our conjecture is that the high error rate is related to the fact of the Matrix Approach be



I) Group Size = 2

II) Group Size = 3

III) Group Size = 7

IV) Group Size = 60

Figure 5.6: Error Representation for coupled spheroid. Error color scale is 0 (blue) to 7.026×10^{-1} (red) meters

based on linear interpolation functions as NMSM uses high-order polynomial.. Therefore, we conducted a test on the square membrane and cube objects because only they can use linear NURBS basis functions since their geometries has no curvature. In this case, the square membrane control mesh is defined with 6×6 control points, cube control mesh with $6 \times 6 \times 6$ control points and we apply linear basis functions in expression (2.24) . The percentage of root mean square error $\%e_{rms}$, and the percentage of maximum error $\%e_{max}$ are shown in Table 5.7 indicating a superiority of our solution (Algorithm 5.1) against the Matrix Approach. Also, by comparing results in Tables 5.6 and 5.7 it is evident that our method is quite stable to the change in the polynomial order.

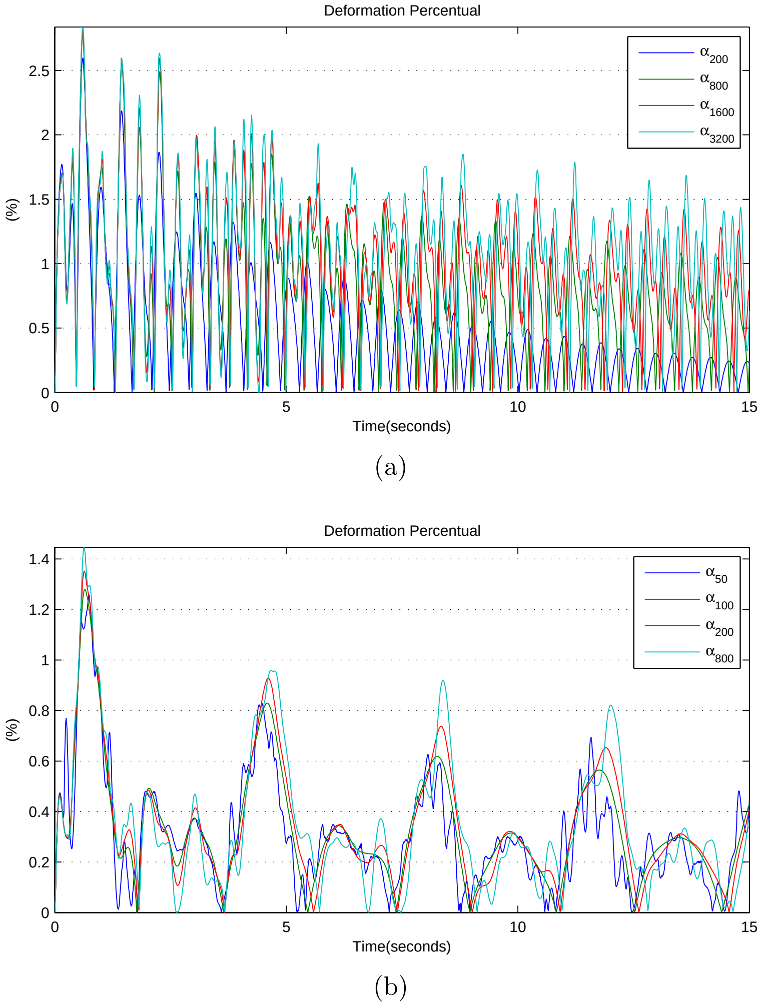
Object	NMSM		Matrix Approach	
	$\%e_{rms}$	$\%e_{max}$	$\%e_{rms}$	$\%e_{max}$
Square	1.19	2.55	9.86	17.73
Cube	2.34	11.33	18.63	28.65

Table 5.7: Comparative under linear basis functions

Experiment V) *Damping and Dynamic Evolution.* In this experiment, we have interest in validating the derived NMSM in time evolution. Also, it is important to study the α parameter effect in expression (4.18) which is used to magnify damping influence. To clarify α effect while validate the NMSM time evolution, experiments were performed to analyze the influence of α on damping in the temporal evolution. To achieve this purpose, we choose the derived NMSM from experiment I) which have some heterogeneity in the spring rest length and minimum error. For 2D case we selected the circular elastic membrane, and coupled spheroid was chosen for 3D case. We kept the same parametrization shown in Table 5.1. So, we instantiate a NMSM by setting up the model parameters with the results generated in the derivation process. The spring coefficients and size of springs groups used were reported in Table 5.2. To compute the masses m_i and damping coefficients c_{ij} we apply the expressions (3.12) and (4.18), respectively.

For circular elastic membrane, we kept time step constant of $\triangle t = {}^1\!/_{240}$ and performed simulations by using a numerical approach based on Leapfrog method for time integration of the ordinary differential equations while for the coupled spheroid we used time step constant of $\triangle t = {}^1\!/_{120}$.

Figure 5.7(a) shows deformation perceptual in time for $\alpha = 200, 800, 1600, 3200$ for the circular membrane, while Figure 5.7(b) shows deformation perceptual in time for $\alpha = 50, 100, 200, 800$ for the coupled spheroid. Both objects were submitted to a small perturbation (step force) during 1/30 initial seconds. We note that the behavior of deformation acquires bumpy contours for all values except for $\alpha = 200$. We also noticed that $\alpha < 200$ causes numerical instability in the simulation of the circular membrane. In the simulation of coupled spheroid occurs numerical instability with $\alpha < 50$. We can see in both results that the α parameter can reveal which is the damping setting that leads to stable temporal evolution.



(a)

(b)

Figure 5.7: Comparative deformation perceptual in time. a) Circular membrane with $\triangle t = {}^{1}/_{240}$ and $\alpha = 200, 800, 1600, 3200$. b) Coupled spheroid with $\triangle t = {}^{1}/_{120}$ and $\alpha = 50, 100, 200, 800$ for

To demonstrate physical realism provide by the derived NMSM model, we provide two movie of 15 second containing the animation for the temporal integration of circular membrane and coupled spheroid as supplementary material.

5.4 DISCUSSIONS

Our proposal have been tested under different computational setups in previous experiments. The main advantage of our method that can be seen in all tests is the ability to derive spring coefficients from NURBS-based IGA framework.

The proposed methodology outperforms the matrix approach in all the reported computational experiments and achieves good results by using basis functions of high order polynomials, which is improvement in relation to other methods in the literature. Once our model is NURBS based, it inherits all apparatus to mesh and geometry manipulation from CAD. As can be seen in the experiments, these features improve the quality of derivation process, the accuracy of geometry and the modeling of the deformable objects.

Chapter

6

In this chapter the results and contributions of this thesis are discussed, and the perspectives of future related topics of research are presented.

CONCLUSIONS AND PERSPECTIVES

This thesis presented a data-driven approach for obtaining the MSM parameters (mass, spring stiffness coefficient and damping constants) from continuous models that had the improvement of accuracy in the interactive mechanical simulations of deformable objects as main motivation.

The main contribution of this thesis was the proposal of the two new methods to design the MSM, that is, to define their inner parameters (mass, stiffness coefficient and damping constant) and mesh topology. The first one, that we called *acceleration based approach*, computes the MSM parameters from linear elastic FEM, that means, the parameterization happens on scenario of small deformations. The second one, namely *IGA-based approach*, determines parameters of the MSM from the NURBS-based IGA that also follows the linear elasticity assumption.

Both approaches proposed are data-driven methods meaning that the unknowns are calculated by solving an optimization problem where is accomplished a fitting of deformation of the MSM to some reference data. For being data-driven, these methods have a number of interesting features that influenced our decision to use them. For instance, they are free of restrictions on physical parameters (there is no restriction on the values of Poisson's ratio and Young's Modulus); and they are not restricted to a specific reference model. In the context of data-driven methods, our contribution was to propose two new objective functions, one for each optimization problem in each method proposed. The new objective functions proposed are the cost function based on acceleration and the cost function based on displacement.

The first one allowed carry out two others contributions that were independent from static equilibrium strategy and the linearization of elastic force. However, it only demonstrated efficiency with 2D objects and some 3D simple geometries, such as cube and plate. It fails when applying for 3D object of complex geometry or with higher order elements.

The cost function based on displacement was applied with the IGA-based approach. The results generated from the application of this cost function performs better than other approaches for all tests conducted in this thesis. Unlike the cost function based on acceleration it showed good results even with complex geometries.

Up to our actual knowledge of the MSM literature, a proposal that combines IGA, NURBS and MSM is new. Particularly, in this thesis was presented an proposal to derive the MSM parameters by using isogeometric analysis. Besides, we also propose a NURBS-based MSM wherein control points are treated like mass points connected by massless springs and the parameterization is accomplished by using the NURBS-based IGA as reference model. Here, we have improvements into two components of a engine to perform the behavior simulation of deformable objects: geometric model and dynamic model. Assuming that the geometric model of the MSM is now a NURBS parameterization means make available for the MSM the great flexibility and precision to represent different shapes including those of most difficult representation as curved ones. Thereat, we carry out more one contribution that is an improvement into accuracy of geometry description applicable to interactive mechanic simulations.

On the other hand, the dynamic model of the MSM is controlled by the model parameters that are mass, stiffness coefficients and damping constants that during the MSM parameterization are assigned by using NURBS-based IGA as reference model. In the NURBS based IGA model, the NURBS basis offer mathematical machinery to get high and easily controlled continuity in the derivatives of basis functions. This fact affects the smoothness of the computed solutions, and, consequently, the precision of the solution. In other words, this high continuity means improving the derivation process and therefore increase the accuracy of the dynamic model MSM.

The NURBS high continuity property makes simple the using of the FEM with higher order elements. It is known that the MSM parameterization from a continuous model with higher order elements has not been explored. Thus, our proposal to parametrize the NMSM by using IGA is also an important contribution that represents a new application of the higher order elements for the MSM parameterization.

We also propose an effective heuristic to deal with trade off between geometric complexity and error rate in optimization problems involving spring parameters (stiffness coefficient, rest length). Our heuristic to cluster the springs demonstrated be an important tool in managing the number variables during the optimization process thereby reducing the high computational cost. Besides, it enabled the location of the best configuration parameters which leads to minimum error state, especially in the parameterization of objects with highly complex geometry.

To develop this thesis we face the need to test and validate the proposals in an environment that could integrate IGA, FEM and MSM algorithms. So, by following the GeoPDEs motivation (see (FALCO; REALI; VáZQUEZ, 2011) for more details) we implement a software framework to serve as an entry point for the practical issues that implementing an IGA code; to be used as a rapid prototyping and testing tool for new IGA algorithms; to allow simple communication between IGA, FEM and MSM models. Similarly to GeoPDEs, our implementation was carried out in MATLAB, but we apply concepts and techniques of object-oriented programming that can increase its flexibility and reduce the maintenance cost. For simplicity, we dubbed our framework as the Multi-model Analysis Framework, abbreviated by MAF. The Multi-model Analysis Framework is described in Appendix A.

6.1 FUTURE WORKS

The MSM has been an attractive alternative to the continuous models for interactive mechanics simulations. In this context, we will describe the future directions of research related to study presented in this thesis.

The 2D MSM models derived from the acceleration based approach demonstrated a suitable accuracy when compared with the corresponding FEM reference models. In particularly, it fails to parameterize the MSM by using higher order elements or complex 3D objects. A research direction within this topic would be to improve the methodology for 3D cases by testing new hypotheses for selection of dominant term. We believe that this point is the source of errors that avoids the convergence to a suitable solution.

The visual realism provided by our NMSM was tested in a few experiments. In all cases we used the MAF framework that is implemented in the MATLAB environment. A fundamental step for this research is to apply the NMSM into real environments of interactive mechanic simulations. In the near future we must use the NMSM into a game engine in order to carry out a further evaluation of how much is gained with the NURBS representation in the MSM.

Our results outperform the matrix approach from the authors (NATSUPARKPONG, 2009; OTAMENDI, 2011). We demonstrated that the matrix approach improves the results when using linear NURBS basis functions. Therefore, a possibility for future research is investigate the matrix approach when considering higher order elements.

The tensor product structure of NURBS makes it harder to perform local mesh refinement. The development of local refinement strategies within IGA is a subject of active research. Local refinement techniques include T-splines, T-meshes and hierarchical B-splines (BAZILEVS et al., 2010; DIMITRI et al., 2014). We plan to improve our approach for NMSM parameterization by including local refinement techniques to be considered during the parameterization process. Also, we intend to apply the local refinement techniques, in the context of online remeshing (PALOC; FARACI; BELLO, 2006), aiming to improve visual realism during real time simulations.

Complex geometry of arbitrary shapes can only be represented by multiple NURBS patches. So, in future we must add multiple NURBS patches in the IGA-based approach since our proposals were evaluated only with single NURBS patches.

Finally, in this thesis we have already highlighted that the parameterization of MSM was performed using linear continuous models as reference ones. Therefore, a very important future work is to parameterize the MSM from a continuous non-linear model.

BIBLIOGRAPHY

AKENINE-MöLLER, T.; HAINES, E.; HOFFMAN, N. *Real-Time Rendering, Third Edition.* [S.l.]: CRC Press, 2008. ISBN 978-1-4398-6529-3.

BASAFA, E.; FARAHMAND, F.; VOSSOUGHI, G. A non-linear mass-spring model for more realistic and efficient simulation of soft tissues surgery. *Stud Health Technol Inform*, v. 132, p. 23–5, 2008.

BATHE, K. *Finite element procedures.* [S.l.]: Prentice Hall, 1996. ISBN 978-0-13-301458-7.

BAUDET, V. *Modélisation et simulation paramétrable d'objets déformables. Application aux traitements des cancers pulmonaires.* Tese (Thèse de Doctorat en Informatique) — Université Lyon1, jun. 2006.

BAUDET, V. et al. *Integrating Tensile Parameters in Mass-Spring System for Deformable Object Simulation.* [S.l.], 2009. 14 p.

BAZILEVS, Y. et al. Isogeometric analysis of Lagrangian hydrodynamics. *Journal of Computational Physics*, v. 243, p. 224 – 243, 2013. ISSN 0021-9991.

BAZILEVS, Y. et al. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, v. 199, n. 5-8, p. 229 – 263, 2010. ISSN 0045-7825.

BENSON, D. J. et al. A generalized finite element formulation for arbitrary basis functions: From isogeometric analysis to XFEM. *International Journal for Numerical Methods in Engineering*, v. 83, n. 6, p. 765–785, 2010. ISSN 1097-0207.

BERTSIMAS, D.; TSITSIKLIS, J. Simulated Annealing. *Statistical Science*, v. 8, n. 1, p. 10–15, 1993.

BIANCHI, G. et al. Simultaneous Topology and Stiffness Identification for Mass-Spring Models Based on FEM Reference Deformations. In: *In MICCAI(2004).* [S.l.: s.n.], 2004. p. 293–301.

BOOR, C. D. *A Practical Guide to Splines.* [S.l.]: Springer-Verlag, 1978. (Applied Mathematical Sciences, vol. 27). ISBN 978-3-540-90356-7.

BOURGUIGNON, D.; CANI, M.-P. Controlling Anisotropy in Mass-Spring Systems. In: MAGNENAT-THALMANN, N.; THALMANN, D.; ARNALDI, B. (Ed.). *Computer Animation and Simulation 2000.* [S.l.]: Springer Vienna, 2000, (Eurographics). p. 113–123. ISBN 978-3-211-83549-4.

CHEN, F. et al. *Soft Tissue Modeling using Nonlinear Mass Spring and Simplified Medial Representation.* [S.l.: s.n.], 2007.

CHOI, K.-S. et al. Deformable simulation using force propagation model with finite element optimization. *Computers & Graphics*, v. 28, n. 4, p. 559 – 568, 2004. ISSN 0097-8493.

COMAS, O. *Real-time Soft Tissue Modelling on GPU for Medical Simulation.* Tese (Doutorado) — INRIA Lille, France, dez. 2010.

COOK, R. *Concepts and applications of finite element analysis.* [S.l.]: Wiley, 2001. ISBN 978-0-471-35605-9.

COTTRELL, J.; HUGHES, T.; BAZILEVS, Y. *Isogeometric Analysis: Toward Integration of CAD and FEA.* [S.l.]: Wiley, 2009. ISBN 978-0-470-74909-8.

DELINGETTE, H. Triangular Springs for Modeling Nonlinear Membranes. *Visualization and Computer Graphics, IEEE Transactions on*, v. 14, n. 2, p. 329–341, 2008. ISSN 1077-2626.

DEUSSEN, O.; KOBBELT, L.; TüCKE, P. Using Simulated Annealing to Obtain Good Nodal Approximations of Deformable Bodies. In: *In Sixth Eurographics Workshop on Simulation and Animation.* [S.l.]: Springer, 1995. p. 30–43.

DIMITRI, R. et al. NURBS- and T-spline-based isogeometric cohesive zone modeling of interface debonding. *Computational Mechanics*, v. 54, n. 2, p. 369–388, 2014. ISSN 0178-7675.

DUKKIPATI, R. *MATLAB for Mechanical Engineers.* [S.l.]: New Age Science, 2009. ISBN 978-1-906574-13-0.

ERLEBEN, K. et al. *Physics-based Animation (Graphics Series).* Rockland, MA, USA: Charles River Media, Inc., 2005. ISBN 1-58450-380-7.

ESPATH, L. F. R.; BRAUN, A. L.; AWRUCH, A. M. An Introduction to Isogeometric Analysis Applied to Solid Mechanics. *Mecanica Computacional*, XXX, n. 1-4, p. 1955–1975, nov. 2011.

ETZMUSS, O.; GROSS, J.; STRASSER, W. Deriving a particle system from continuum mechanics for the animation of deformable objects. *Visualization and Computer Graphics, IEEE Transactions on*, v. 9, n. 4, p. 538–550, out. 2003. ISSN 1077-2626.

FALCO, C. d.; REALI, A.; VáZQUEZ, R. GeoPDEs: A research tool for Isogeometric Analysis of {PDEs}. *Advances in Engineering Software*, v. 42, n. 12, p. 1020 – 1034, 2011. ISSN 0965-9978.

FARIN, G. *Curves and surfaces for computer-aided geometric design: a practical guide.* [S.l.]: Academic Press, 1997. (Computer science and scientific computing, vol. 1). ISBN 978-0-12-249054-5.

FRANK, B. et al. Learning object deformation models for robot motion planning. *Robotics and Autonomous Systems*, v. 62, n. 8, p. 1153 – 1174, 2014. ISSN 0921-8890.

GELDER, A. V. Approximate simulation of elastic membranes by triangulated spring meshes. *J. Graph. Tools*, v. 3, n. 2, p. 21–42, fev. 1998. ISSN 1086-7651.

GIRALDI, G. A.; ORTIZ, J. S. E.; JR, J. M. P. Modelos Massa-Mola para Visualizacao de Tecidos Flexíveis. *Revista Eletrônica de Iniciação Científica*, v. 6, n. 2, p. 16, jun. 2006.

HENG, B. C. P.; MACKIE, R. I. Using design patterns in object-oriented finite element programming. *Computers & Structures*, v. 87, n. 15–16, p. 952 – 961, 2009. ISSN 0045-7949. Computational Structures Technology.

HORáK, M.; PATZáK, B.; JIRáSEK, M. On design of element evaluators in OOFEM. *Advances in Engineering Software*, v. 72, n. 0, p. 193 – 202, 2014. ISSN 0965-9978. Special Issue dedicated to Professor Zdeněk Bittnar on the occasion of his Seventieth Birthday: Part 2.

HUANGFU, Z.; YAN, L.; LIU, X. An Improved Mass-spring Model for Simulation of Soft Tissue Deformation. *Journal of Information and Computational Science*, v. 10, n. 17, nov. 2013.

HUGHES, T. J. R.; COTTRELL, J. A.; BAZILEVS, Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, v. 194, n. 39-41, p. 4135–4195, 2005. ISSN 0045-7825.

IX, F. D.; QIN, H.; KAUFMAN, A. A novel haptics-based interface and sculpting system for physics-based geometric design. *Computer-Aided Design*, v. 33, n. 5, p. 403 – 420, 2001. ISSN 0010-4485.

JACOBSON, A. et al. Skinning: Real-time Shape Deformation. In: *ACM SIGGRAPH 2014 Courses*. [S.l.: s.n.], 2014.

JAMES, D. L.; PAI, D. K. ArtDefo: Accurate Real Time Deformable Objects. In: *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999. (SIGGRAPH '99), p. 65–72. ISBN 0-201-48560-5.

JARAMILLO, A.; PRIETO, F.; BOULANGER, P. Deformable part inspection using a spring–mass system. *Computer-Aided Design*, v. 45, n. 8–9, p. 1128 – 1137, 2013. ISSN 0010-4485.

KELLY, S. *Mechanical Vibrations: Theory and Applications*. [S.l.]: Cengage Learning, 2011. ISBN 978-1-133-41900-6.

KäHLER, K.; HABER, J.; SEIDEL, H.-P. Geometry-based Muscle Modeling for Facial Animation. In: *Proceedings of Graphics Interface 2001*. Toronto, Ont., Canada, Canada: Canadian Information Processing Society, 2001. (GI '01), p. 37–46. ISBN 0-9688808-0-0.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by Simulated Annealing. *Science*, v. 220, n. 4598, p. 671–680, 1983.

LEE, D. et al. *Modeling and simulation of skeletal muscle for computer graphics: A survey*. [S.l.]: Now, 2012.

LEE, Y.; TERZOPOULOS, D.; WATERS, K. Realistic modeling for facial animation. In: *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1995. (SIGGRAPH '95), p. 55–62. ISBN 0-89791-701-4.

LIU, T. et al. Fast Simulation of Mass-Spring Systems. *ACM Transactions on Graphics*, v. 32, n. 6, p. 209:1–7, nov. 2013. Proceedings of ACM SIGGRAPH Asia 2013, Hong Kong.

LLOYD, B. A. et al. Identification of Dynamic Mass Spring Parameters for Deformable Body Simulation. In: MANIA, K.; REINHARD, E. (Ed.). *Eurographics 2008 - Short Papers*. [S.l.: s.n.], 2008. p. 131–134.

LLOYD, B. A.; SZéKELY, G.; HARDERS, M. Identification of Spring Parameters for Deformable Object Simulation. *Visualization and Computer Graphics, IEEE Transactions on*, v. 13, n. 5, p. 1081–1094, set. 2007.

LOGAN, D. *A First Course in the Finite Element Method*. [S.l.]: Nelson Education Limited, 2011. ISBN 978-0-495-66825-1.

LONG, J.; BURNS, K.; YANG, J. Cloth Modeling and Simulation: A Literature Survey. In: DUFFY, V. (Ed.). *Digital Human Modeling*. [S.l.]: Springer Berlin Heidelberg, 2011, (Lecture Notes in Computer Science, v. 6777). p. 312–320. ISBN 978-3-642-21798-2.

MACKERLE, J. Object-oriented Programming in FEM and BEM: A Bibliography (1990-2003). *Adv. Eng. Softw.*, v. 35, n. 6, p. 325–336, jun. 2004. ISSN 0965-9978.

MCKENNA, F.; SCOTT, M. H.; FENVES, G. L. Nonlinear Finite-Element Analysis Software Architecture Using Object Composition. *Journal of Computing in Civil Engineering*, v. 24, n. 1, p. 95–107, 2010.

MEIER, U. et al. Real-time deformable models for surgery simulation: a survey. *Computer Methods and Programs in Biomedicine*, v. 77, n. 3, p. 183–197, 2005.

METROPOLIS, N. et al. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, v. 21, n. 6, p. 1087–1092, 1953.

MOORE, P.; MOLLOY, D. A Survey of Computer-Based Deformable Models. In: *Machine Vision and Image Processing Conference, 2007. IMVIP 2007. International.* [S.l.: s.n.], 2007. p. 55–66.

MORRIS, D.; SALISBURY, K. *Automatic Preparation, Calibration, and Simulation of Deformable Objects.* [S.l.], 2007.

NATSUPAKPONG, S.; ÇAVUSOGLU, M. C. Determination of elasticity parameters in lumped element (mass-spring) models of deformable objects. *Graph. Models*, v. 72, n. 6, p. 61–73, nov. 2010. ISSN 1524-0703.

NATSUPARKPONG, S. *Physically Based Modeling and Simulation for Virtual Environment Based Surgical Training.* Tese (Doutorado) — Case Western Reserve University, United State, jul. 2009.

NEALEN, A. et al. Physically Based Deformable Models in Computer Graphics. *Computer Graphics Forum*, v. 25, n. 4, p. 809–836, 2006.

NGUYEN, V.; BORDAS, S. Extended Isogeometric Analysis for Strong and Weak Discontinuities. In: BEER, G.; BORDAS, S. (Ed.). *Isogeometric Methods for Numerical Simulation.* [S.l.]: Springer Vienna, 2015, (CISM International Centre for Mechanical Sciences, v. 561). p. 21–120. ISBN 978-3-7091-1842-9.

OTAMENDI, G. S. V. *Designing deformable models of soft tissue for virtual surgery planning and simulation using the Mass-Spring Model.* Tese (Doutorado) — University of Navarra, Spain, jan. 2011.

PALOC, C. et al. Online Multiresolution Volumetric Mass Spring Model for Real Time Soft Tissue Deformation. In: DOHI, T.; KIKINIS, R. (Ed.). *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2002.* [S.l.]: Springer Berlin Heidelberg, 2002, (Lecture Notes in Computer Science, v. 2489). p. 219–226. ISBN 978-3-540-44225-7.

PALOC, C.; FARACI, A.; BELLO, F. Online Remeshing for Soft Tissue Simulation in Surgical Training. *Computer Graphics and Applications, IEEE*, v. 26, n. 6, p. 24–34, nov. 2006. ISSN 0272-1716.

PIEGL, L. On NURBS: A Survey. *IEEE Comput. Graph. Appl.*, v. 11, n. 1, p. 55–71, jan. 1991. ISSN 0272-1716.

PIEGL, L.; TILLER, L. *The Nurbs Book.* [S.l.]: Springer-Verlag GmbH, 1997. (Monographs in Visual Communication Series). ISBN 978-3-540-61545-3.

POZRIKIDIS, C. *Numerical Computation in Science and Engineering.* [S.l.]: Oxford University Press, 2008. (Numerical Computation in Science and Engineering). ISBN 978-0-19-537611-1.

PROVOT, X. Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior. 1996.

QUEK, S.; LIU, G. *Finite Element Method: A Practical Course: A Practical Course.* [S.l.]: Elsevier Science, 2003. (Referex Engineering). ISBN 978-0-7506-5866-9.

RAO, S. S. *The Finite Element Method in Engineering.* [S.l.]: Elsevier Science, 2004. ISBN 978-0-7506-7828-5.

REDDY, J. *An Introduction to the Finite Element Method.* [S.l.]: McGraw-Hill Companies,Incorporated, 2005. (McGraw-Hill Mechanical Engineering). ISBN 978-0-07-246685-0.

REDDY, J. *An Introduction to Continuum Mechanics.* [S.l.]: Cambridge University Press, 2013. ISBN 978-1-107-29240-6.

ROYER, L. et al. Real-time Tracking of Deformable Target in 3d Ultrasound Images. In: *IEEE Int. Conf. on Robotics and Automation, ICRA'15.* Seattle, United States: [s.n.], 2015.

SADD, M. *Elasticity: Theory, Applications, and Numerics.* [S.l.]: Elsevier Science, 2009. ISBN 978-0-08-092241-6.

SALA, A. et al. Integration of biomechanical parameters in tetrahedral mass-spring models for virtual surgery simulation. In: *Engineering in Medicine and Biology Society,EMBC, 2011 Annual International Conference of the IEEE.* [S.l.: s.n.], 2011. p. 4550–4554.

SAN-VICENTE, G.; AGUINAGA, I.; CELIGUETA, J. T. Cubical Mass-Spring Model Design Based on a Tensile Deformation Test and Nonlinear Material Model. *IEEE Transactions on Visualization and Computer Graphics*, v. 18, n. 2, p. 228–241, fev. 2012. ISSN 1077-2626.

SILVA, J. P. d.; GIRALDI, G. A.; APOLINARIO, A. L. A new optimization approach for mass-spring models parameterization. *Graphical Models*, v. 81, p. 1 – 17, set. 2015. ISSN 1524-0703.

SPENCER, A. *Continuum Mechanics.* [S.l.]: Dover Publications, 2004. (Dover books on physics). ISBN 978-0-486-43594-7.

TEMIZER, .; WRIGGERS, P.; HUGHES, T. J. R. Contact treatment in isogeometric analysis with {NURBS}. *Computer Methods in Applied Mechanics and Engineering*, v. 200, n. 9–12, p. 1100 – 1112, 2011. ISSN 0045-7825.

TERZOPOULOS, D. et al. Elastically deformable models. *SIGGRAPH Comput. Graph.*, v. 21, n. 4, p. 205–214, ago. 1987. ISSN 0097-8930.

TERZOPOULOS, D.; QIN, H. Dynamic NURBS with geometric constraints for interactive sculpting. *ACM Trans. Graph.*, v. 13, n. 2, p. 103–136, abr. 1994. ISSN 0730-0301.

VABLE, M. *Mechanics of Materials.* [S.l.]: Oxford University Press, 2009. ISBN 978-0-19-513337-0.

VANGELDER, A.; WILHELMS, J. *Simulation of Elastic Membranes and Soft Tissue With Triangulated Spring Meshes.* Santa Cruz, CA, USA, 1997.

VICENTE, G. S. et al. Maxillofacial surgery simulation using a mass-spring model derived from continuum and the scaled displacement method. *International Journal of Computer Assisted Radiology and Surgery*, v. 4, n. 1, p. 89–98, 2009. ISSN 1861-6410. 10.1007/s11548-008-0271-0.

VOLINO, P.; THALMANN, N. *Virtual Clothing.: Theory and Practice.* [S.l.]: Springer-Verlag GmbH, 2000. ISBN 978-3-540-67600-3.

VOLLINGER, U. et al. Evolutionary optimization of mass-spring models. *CIRP Journal of Manufacturing Science and Technology*, v. 1, n. 3, p. 137 –141, 2009. ISSN 1755-5817. \textlessce:title\textgreaterDesign Synthesis\textless/ce:title\textgreater.

WU, J.; WESTERMANN, R.; DICK, C. A Survey of Physically Based Simulation of Cuts in Deformable Bodies. *Computer Graphics Forum*, p. n/a–n/a, 2015. ISSN 1467-8659.

XU, S. et al. A Nonlinear Viscoelastic Tensor-Mass Visual Model for Surgery Simulation. *Instrumentation and Measurement, IEEE Transactions on*, v. 60, n. 1, p. 14 –20, jan. 2011. ISSN 0018-9456.

YAGHOOBI, A. Implementation of Nonlinear Finite Element Using Object–Oriented. *Journal of Mechanical Engineering and Technology*, p. 61–79, jun. 2012. ISSN 2180-1053.

YUAN, Z.; FISH, J. Nonlinear multiphysics finite element code architecture in object oriented Fortran environment. *Finite Elements in Analysis and Design*, v. 99, p. 1 – 15, 2015. ISSN 0168-874X.

ZERBATO, D.; GALVAN, S.; FIORINI, P. Calibration of mass spring models for organ simulations. In: *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on.* [S.l.: s.n.], 2007. p. 370–375.

ZIENKIEWICZ, O.; TAYLOR, R.; ZHU, J. *The Finite Element Method: Its Basis And Fundamentals.* [S.l.]: Elsevier Butterworth-Heinemann, 2005. (The Finite Element Method). ISBN 978-0-7506-6320-5.

Appendix A

To develop this doctoral work we face the need to test and validate our proposal in an environment that integrates IGA, FEM and MSM algorithms. So, by following the GeoPDEs (see (FALCO; REALI; VáZQUEZ, 2011) for more details) motivation we implement a framework to serve as an entry point for the practical issues that implementing an IGA code; to be used as a rapid prototyping and testing tool for new IGA algorithms; to allow simple communication between IGA, FEM and MSM models. Similarly to GeoPDEs, our implementation was carried out in MATLAB, but we apply concepts and techniques of object-oriented programming that can increase its flexibility and reduce the maintenance cost. For simplicity, we dubbed our framework as the Multi-model Analysis Framework, abbreviated by MAF.

MULTI-MODEL ANALYSIS FRAMEWORK

There are several approaches to implement the finite element method by applying the object-oriented paradigm. These approaches explore oriented-object features as efficiency, flexibility, extensibility, reusability, modularity and portability to generate scalable implementations that can be used into different classes of problems with a minimum effort. Lot of these approaches implement a separation between the physical model and the analysis process. (MACKERLE, 2004; HENG; MACKIE, 2009; MCKENNA; SCOTT; FENVES, 2010; YAGHOOBI, 2012; HORáK; PATZáK; JIRáSEK, 2014; YUAN; FISH, 2015). This separation encapsulates the numerical objects and algorithms of the analysis into an analysis container and the FEM components as elements, nodes, constraints into a model container. We follow a similar reasoning, however, unlike the approaches FEM which are concerned with a single model our architecture must fulfill requirements to multiple models since we must deal with FEM, MSM, IGA and NMSM. Therefore, we choose to distribute the analysis activities between the components of model.

The package diagram in Figure A.1 shows an overview of the components in our arquitecture. In this diagram that basically uses the UML aggregation relationship, each package is actually a set of classes responsible for a series of functionalities. It is noted that the *Model* package is the great aggregator of the MAF. It gathers the functionalities from the *Mesh* and *Integrator* packages to perform the analysis on the dynamic and static scenario. The package *Integrator* implements numerical algorithms for time integration while the *Mesh* package provides through an inheritance chain all components of models as elements, nodes, constraints, springs, material properties, connectivity elements, degree

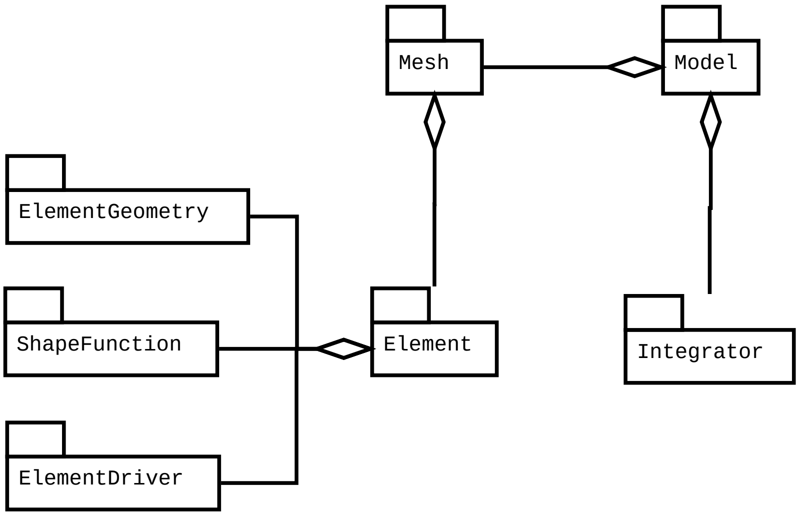


Figure A.1: Packages diagram in UML for components of the Multi-Model Analysis Framework (MAF). One can see the UML aggregation relationship represented as a hollow diamond shape on the containing package with a single line that connects it to the contained package..

of freedom etc. The package *Element* contained in the package *Mesh* is fundamental for the whole architecture. It defines the element geometry, shape functions and element drivers. The element geometry relates to the number of nodes, edges and faces belonging to the element. The shape functions is set of classes to provide data and values about interpolation functions. The package *Element Driver* is responsible for implementing the calculation of matrices stiffness, mass, damping and load vector at element level under different schemes of analysis as plane stress, plane strain. Next, we detail each of these packages.

The package *Model* contains six classes: *DeformableModel*, *FEModel*, *IGAModel*, *ParticleModel*, *MSModel* and *NMSModel*. The UML class diagram in Figure A.2(a) shows the association among these classes. We can see that the *DeformableModel* class is a common ancestral to the other classes which through inheritance mechanism they implement the specificities of each model. The *modelMesh* attribute of *DeformableModel* class is the gateway to the model characteristics. It is assigned internally during the instancing process which depending on the model (FEM, IGA, MSM or NMSM) is created a particular mesh for that model. Furthermore, it also provides attributes and operations of the element such as matrices computation (stiffness, mass and damping) and load vector valuation which are used in global assembling performed by method *AssemblyMatrix*. The implementation of this method by the *MSModel* class occurs only to the particular case of the MSM linearized, where we apply Equation (3.10) of Chapter 2.

The package *Model* keeps closely connection with the package *Mesh*. For each model there is a mesh whose attributes and operations are designed to address the requirements of corresponding model. This means there is a one-to-one relationship between model and mesh, ie the *FEModel* class has its corresponding mesh *MeshFEM*, *IGAModel* class has its corresponding mesh *MeshIGA* and so on. In Figure Figure A.2(b) is shown the mesh classes and as can be seen the association between model and mesh is immediate. The base class in package *Mesh* is the *MeshRoot* class. It contains the definition of geometry (attributes *nodesPos* and *nodesLink*), boundary condition (attribute *Boundaries*), element and operations responsible for configuring the mesh. Also part of this class the attribute igaParam that is defined by classes *MeshIGA* and *MeshNMSM* to provide the data structure of IGA apparatus which gathers NURBS parameterization and connectivity data structures of the IGA elements. Beyond the common characteristics, the class *MeshPring* brings together the operations to management of springs that is the core to the *ParticleModel*, *MSModel* and *NMSModel* models.

Many numerical integration methods are available for the approximate solution of differential equations like those seen in Chapter 2 which describe the mechanical behavior of deformable bodies (the readers interested can find more details about these methods in references (BATHE, 1996; DUKKIPATI, 2009)). In general, numerical integration is used to find the system state which contains positions and velocities of the deformable object from the system of equations defining the model. We supply four these methods in package *Integrator*. The classes *CDMDisplacement* and *CDMParticle* contain central difference method for FEM and particles based models, respectively. The Newmark and Leapfrog method are implemented in the homonym classes.

The package *Element* shown in Figure A.3 contains the hierarchy of classes used to define the element of each model. The geometry of element consisting by the edges and faces is defined on the *ElementBase* class. The instancing of the geometry data is carried out by the classes in the package *ElementGeometry* as can be observed in Figure A.4. For the FEM-based models, the assembly process at element level is implemented by methods *getStiffnessMatrix*, *getMassMatrix*, *getDampingMatrix* and *getForceVector*. These methods keeps a close dependence on the packages *ElementDriver* and *ShapeFunction* shown in Figure A.5 (a) and (b), respectively. Recalling the flowchart 2.3 seen in Chapter 2, we can note that the assembly process is to calculate each integration point (quadrature points) into element structures (matrices stiffness, mass, damping and load vector) and assembled local element structures into global ones. The evaluation of integration point is performed by methods *getKe*, *getMe*, *getDe* and *getFe* from the specialized classes of the package *ElementDriver* in collaboration with the package *ShapeFunction.* These computed points are assembled into local arrays which are returned by methods *getStiffnessMatrix*, *getMassMatrix*, *getDampingMatrix* and *getForceVector.*

To instantiate the MAF objects, we define the *makSetting* class (Figure A.6) that centralizes the instantiation process for each model. It has the method *createModel* which is executed to instantiate an object-model from the package *Model* by using the parametrization kept in the data structure *modelParam.* The method *createREFMSM* is responsible to instantiate a MSM model that is counterpart to model parametrized in *modelParam* (model reference).

The instance of the model is created when we accomplish the following steps:

1. Parametrize the data structure for boundary condition;

2. Parametrize the data structure for model mesh;

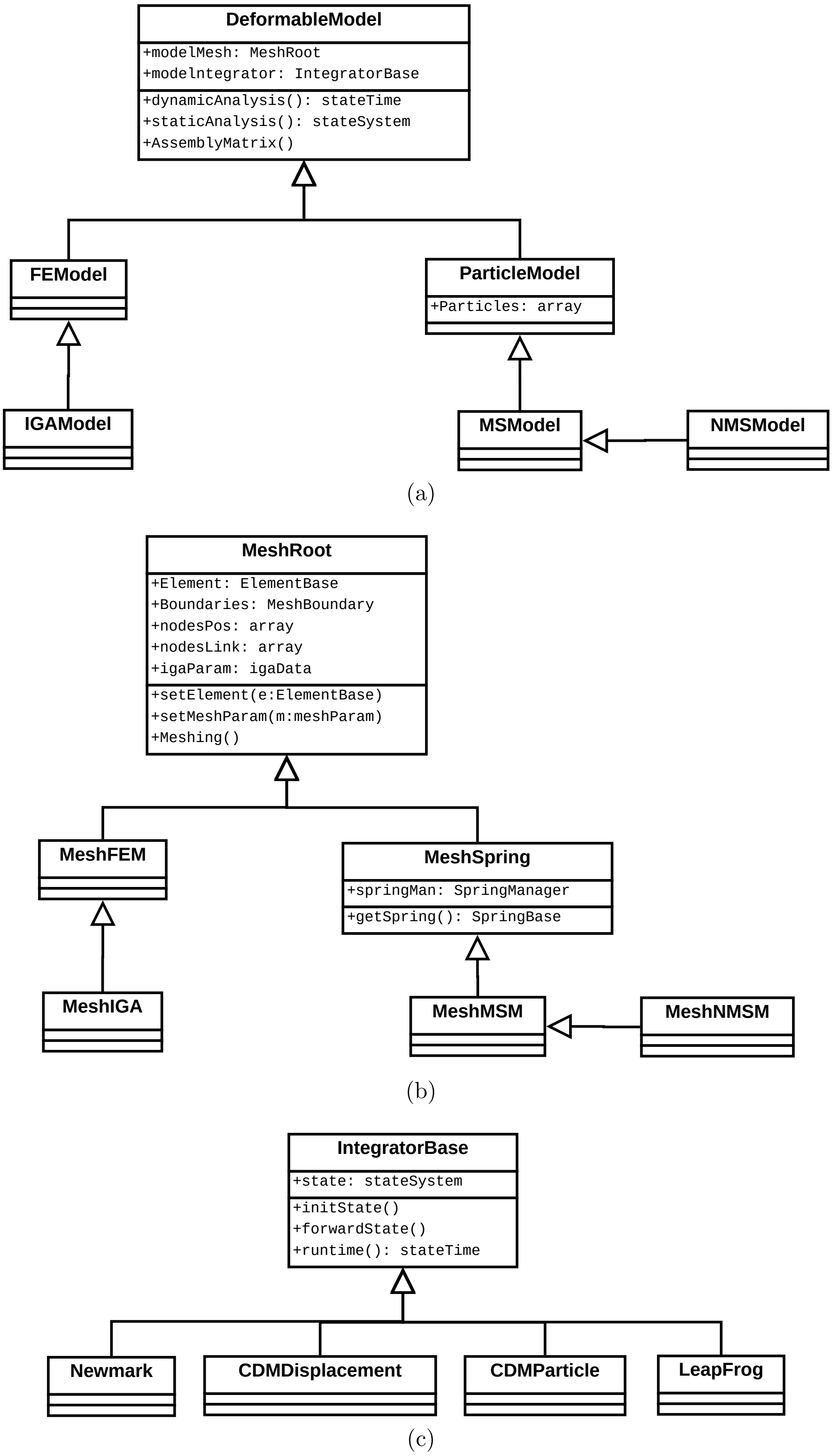


Figure A.2: The UML class diagram to the packages a) *Model*; b) *Mesh*; c) *Integrator*. All classes inherit general characteristics of a) *DeformableModel,* b) *MeshRoot*, c) *IntegratorBase* base classes and they implement the specificities of each package.

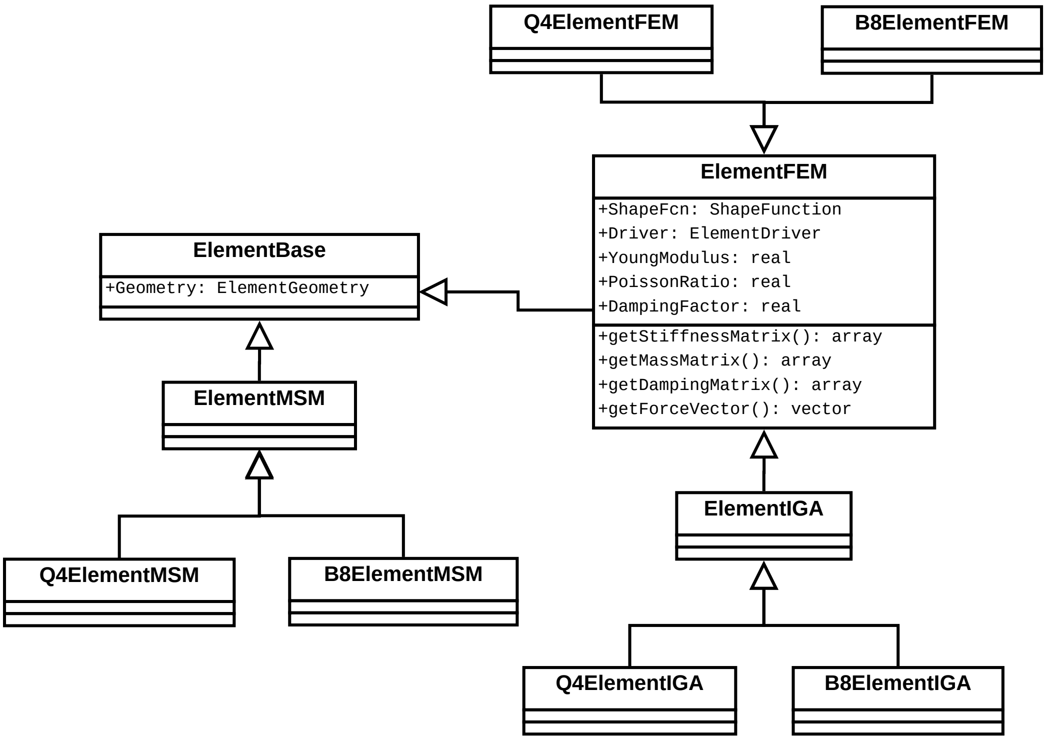


Figure A.3: The UML class diagram to the package *Element*. All classes inherit geometry attributes from base class *ElementBase*. These attributes are actually created by the classes *ElementFEM*, *ElementMSM* and *ElementIGA* observing the specificities of each model. In classes at the final of hierarchy are defined the nodes number, edges, faces, shape function, material properties.

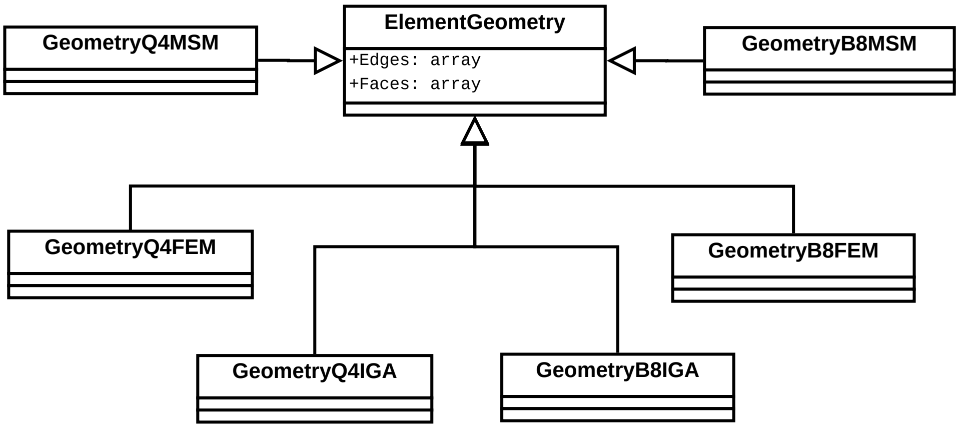


Figure A.4: The UML class diagram to the package *ElementGeometry*. This package just sets the composition of edges and faces to each element of the model on a one-to-one relationship.

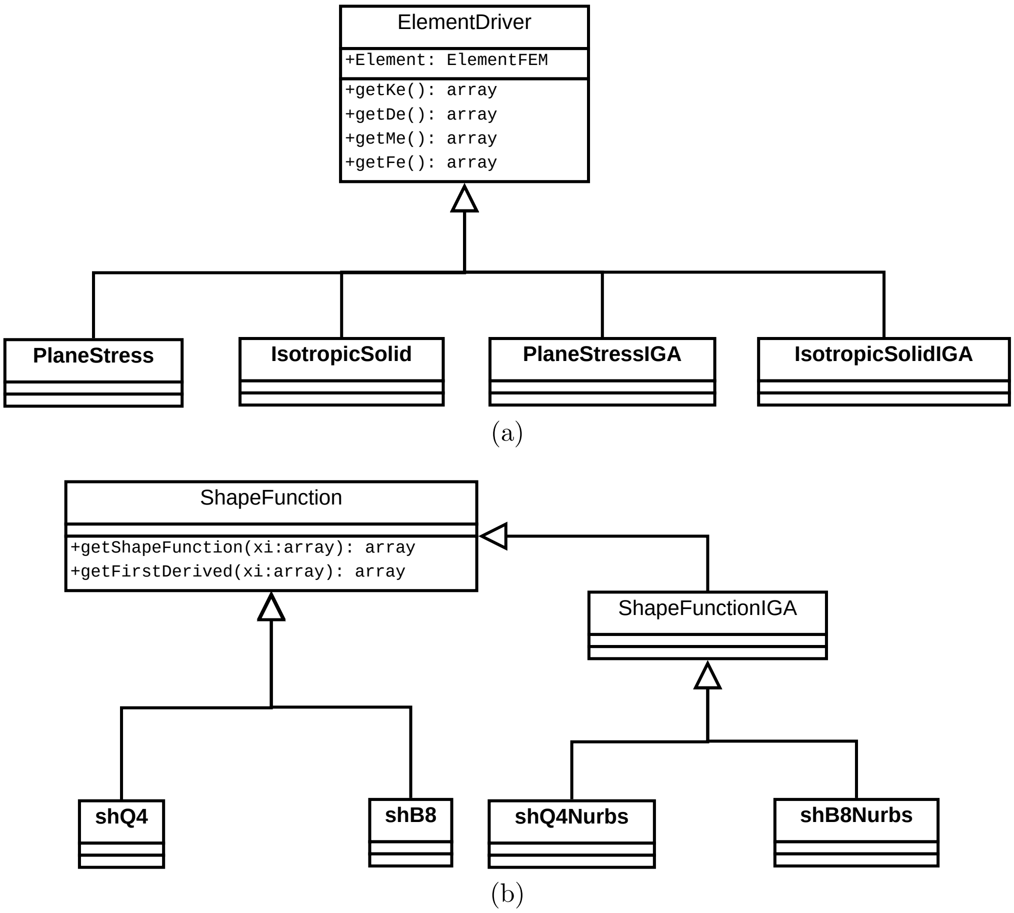


Figure A.5: The UML class diagram to the packages a) *ElementDriver* and b) *ShapeFunction*. *ElementDriver* is an abstract class that provides operations to be implemented by specialized classes in a given analysis scheme. The *ShapeFunction* abstract class provides methods to evaluate specialized interpolation functions (Lagrange Polynomial, NURBS basis functions).

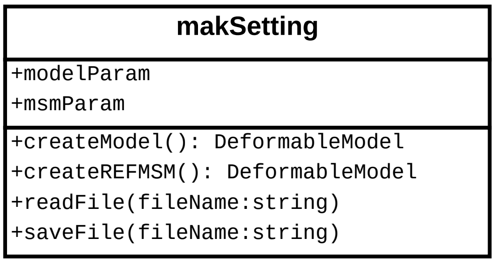

makSetting
+modelParam
+msmParam
+createModel(): DeformableModel
+createREFMSM(): DeformableModel
+readFile(fileName:string)
+saveFile(fileName:string)

Figure A.6: The *makSetting* class with its attibutes and methods. The *modelParam* attribute keeps data structure with the parametrization of model. The *msmParam* attribute keeps the MSM parametrization related with *modelParam*.

3. Parametrize the data structure for physical parameter of model;

4. Parametrize the data structure for model, and if applicable also assign MSM model parameter;

5. Finally, instantiate the model with its MSM counterpart (if applicable).

We provide a MATLAB script by the Algorithm A.1 which shows the instantiation of the *FEModel* class and its MSM counterpart.

Algorithm A.1 The MATLAB script to create an instance of model

```
bcdata = createBC('geometry','bottom',[1;1;1],1);
meshparam = createMeshParam('Q4FEM',{ {'sz',[2 2 0]},...
                                      {'ne',[4 4 0]},...
                                      {'bc',bcdata}});
physparam = createPhysParam({{'elasticity',14630},{'poisson',1/3},...
                             {'density',1144},{'thickness',0.1}});
paramFEM = createModelParam('fem',meshparam,physparam);
paramMSM = defaultMSMParam('Q4MSM',{'stretch','shear'});
cfg = setFEM(paramFEM,'SquareMembrane',paramMSM);
fem = cfg.createModel();
msm = cfg.createREFMSM();
```