Hybrid 3D Mass Spring System for Soft Tissue Simulation

Karolina Golec

THÉSE DE DOCTORAT DE L'UNIVERSITÉ DE LYON
opérée au sein de
L'UNIVERSITÉ CLAUDE BERNARD LYON 1

École Doctorale ED 512
InfoMaths

Spécialité de doctorat: Informatique
Discipline: Modélisation et simulation

Soutenue publiquement le 19/01/2018, par:
Karolina GOLEC

Hybrid 3D Mass Spring System for Soft Tissue Simulation

Devant le jury composé de:

Maud MARCHAL, Maître de Conférences HDR INSA Rennes, IRISA	Rapporteure
Yohan PAYAN, Directeur de Recherche CNRS, TIMC-IMAG	Rapporteur
Laurence CHÈZE, Professeure UCBL, LBMC	Examinatrice
Sébastien LAPORTE, Professeur ENSAM, IBHGC	Examinateur
Philippe MESEURE, Professeur UP, XLIM	Examinateur
Guillaume DAMIAND, Chargé de Recherche HDR CNRS, LIRIS	Directeur de thèse
Florence ZARA, Maître de Conférences UCBL, LIRIS	Co-directrice de thèse
Stéphane NICOLLE, Maître de Conférences UCBL, LBMC	Co-directeur de thèse
Jean-François PALIERNE, Chargé de Recherche CNRS, Lab. Physique, ENS de Lyon	Invité

April 9, 2018

"You will never feel truly satisfied by work until you are satisfied by life."

Heather Schuck, The Working Mom Manifesto

Resumé

La nécessité de simulations de tissus mous, tels que les organes internes, se pose avec le progrès des domaines scientifiques et médicaux. Le but de ma thèse est de développer un nouveau modèle générique, topologique et physique, pour simuler les organes humains. Un tel modèle doit être facile à utiliser, doit pouvoir effectuer des simulations en temps réel avec un niveau de précision permettant l'utilisation à des fins médicales.

Cette thèse explore de nouvelles méthodes de simulation et propose des améliorations pour la modélisation de corps déformables. Les méthodes proposées visent à pouvoir effectuer des simulations rapides, robustes et fournissant des résultats physiquement précis. L'intérêt principal de nos solutions réside dans la simulation de tissus mous élastiques à petites et grandes déformations à des fins médicales. Nous montrons que pour les méthodes existantes, la précision pour simuler librement des corps déformables ne va pas de pair avec la performance en temps de calcul. De plus, pour atteindre l'objectif de simulation rapide, de nombreuses approches déplacent certains calculs dans une étape de pré-traitement, ce qui entraîne l'impossibilité d'effectuer des opérations de modification topologiques au cours de la simulation comme la découpe ou le raffinement.

Dans cette thèse, le cadre utilisé pour les simulations s'appelle TopoSim. Il est conçu pour simuler des matériaux à l'aide de systèmes masses-ressorts (MSS) avec des paramètres d'entrée spécifiques. En utilisant un MSS, qui est connu pour sa simplicité et sa capacité à effectuer des simulations temps réel, nous présentons plusieurs améliorations basé physiques pour contrôler les fonctionnalités globales du MSS qui jouent un rôle clé dans la simulation de tissus réels.

La première partie de ce travail de thèse vise à reproduire une expérience réelle de simulation physique qui a étudié le comportement du tissu porcin à l'aide d'un rhéomètre rotatif. Son objectif était de modéliser un corps viscoélastique non linéaire. À partir de l'ensemble des données acquises, les auteurs de l'expérience ont dérivé une loi de comportement visco-élastique qui a ensuite été utilisée afin de la comparer avec nos résultats de simulation. Nous définissons une formulation des forces viscoélastiques non linéaires inspirée de la loi de comportement physique. La force elle-même introduit une non-linéarité dans le système car elle dépend fortement de l'amplitude de l'allongement du ressort et de trois paramètres spécifiques à chaque type de tissu.

La seconde partie de la thèse présente notre travail sur les forces de correction de volume permettant de modéliser correctement les changements volumétriques dans un MSS. Ces forces assurent un comportement isotrope des solides élastiques et un comportement correct du volume quelquesoit la valeur du coefficient de Poisson utilisé. La méthode nécessite de résoudre deux problèmes: l'instabilité provoquant des plis et les contraintes de Cauchy. Nos solutions à ces limitations impliquent deux étapes. La première consiste à utiliser trois types de ressorts dans un maillage entièrement hexaédrique: les arêtes, les faces diagonales et les diagonales internes. Les raideurs des ressorts dans le système ont été formulées pour obéir aux lois mécaniques de base. La deuxième étape consiste à ajouter des forces de correction linéaires calculées en fonction du changement de volume et des paramètres mécaniques du tissu simulé, à savoir le coefficient de Poisson et le module de Young.

La troisième partie concerne les aspects de création d'un maillage précis et léger. C'est en effet un des éléments les plus importants lorsqu'il s'agit de simulations de corps réels. Dans ce chapitre, nous présentons une introduction à la mise en œuvre de maillages d'éléments mixtes construits avec les éléments 3D suivants: tétraèdres, prismes, pyramides et hexaèdres. Le maillage mixte nous permet d'obtenir un niveau de détail plus précis tout en diminuant le nombre d'éléments à utiliser. Une telle structure nécessite

une formulation correcte des raideurs des ressorts pour chaque élément, pour produire une méthode physiquement réaliste. Notre cadre générique peut être étendu au maillages mixtes en se basant sur les mêmes lois que la structure entièrement hexaédrique. Le modèle mixte devra ensuite être comparé aux maillages validés existants ainsi qu'aux solutions FEM.

Les systèmes masses-ressorts sont connus pour être simples et faciles à utiliser, mais ils nécessitent une base physique solide pour être utilisés dans le cadre de simulations physiques précises. D'un autre côté, l'environnement médical cherche constamment à pousser les méthodes dans le sens de la simulation en temps réel d'objets déformables détaillés. Les méthodes présentées dans ce travail nous permettent de contrôler globalement la non-linéarité et le volume d'un système masses-ressorts, et améliorent également sa précision. De plus, la mise en œuvre de maillage mixtes facilitera des simulations précises avec des performances en temps réel en diminuant le nombre d'éléments requis.

Abstract

The need for simulations of soft tissues, like internal organs, arises with the progress of the scientific and medical environments. The goal of my PhD is to develop a novel generic topological and physical model to simulate human organs. Such a model shall be easy to use, perform the simulations in the real time and which accuracy will allow usage for the medical purposes.

This thesis explores novel simulation methods and improvement approaches for modeling deformable bodies. The methods aim at fast and robust simulations with physically accurate results. The main interest lies in simulating elastic soft tissues at small and large strains for medical purposes. We show however, that in the existing methods the accuracy to freely simulate deformable bodies and the real-time performance do not go hand in hand. Additionally, to reach the goal of simulating fast, many of the approaches move the necessary calculations to pre-computational part of the simulation, which results in inability to perform topological operations like cutting or refining.

The framework used for simulations in this thesis is called TopoSim and it is designed to simulate materials using Mass Spring Systems (MSS) with particular input parameters. Using Mass-Spring System, which is known for its simplicity and ability to perform fast simulations, we present several physically-based improvements to control global features of MSS which play the key role in simulation of real bodies.

First part of the thesis present work, which aim is to reproduce a real experiment. The experiment studied the behavior of porcine tissue using a rotational rheometer. Its objective was to model non-linear visco-elastic body. From the set of acquired data the authors of the experiment derived a visco-elastic constitutive law which was then used to compare with our simulation results. We define a non-linear visco-elastic force formulation inspired by the constitutive law. The force itself introduces non-linearity in the system as it strongly depends on the magnitude of the spring elongation and three parameters specific for each type of tissue.

Next part presents work on volume correction forces to correctly model the volumetric changes in MSS. It ensures isotropic behavior of elastic solids and correct volume behavior based on a free modification of Poisson's ratio. The method requires us to solve two problems: wrinkle instability and Cauchy's constraints. Our solutions to the presented limitations of MSS involve thus two steps. The first step is to use three types of springs in an all-hexahedral mesh: edge, face-diagonal and inner-diagonal ones. The stiffnesses of the springs in the system were formulated to obey the basic mechanical laws. Next step involved adding the corrections to computed linear forces based on the change of volume and given mechanical parameters of the simulated tissue, namely the Poisson ratio and Young modulus.

Creating a detailed and lightweight mesh structure is one of the most important elements when dealing with simulations of real bodies. Therefore the last chapter presents an introduction to implementation of mixed-element meshes built of the following 3D elements: tetrahedra, prisms, pyramids and hexahedra. The mixed-element mesh allows us to obtain the desired level of detail of a mesh and use the minimal required number of elements in the same time. Such a structure requires a correct formulation of the stiffnesses for particular springs to be integrated as a physically plausible method into our framework. We talk about the solution basis: it shall be based on the same laws as our all-hexahedral structure. The model built of different types of elements shall be then compared to the already existing validated meshes as well as FEM solutions.

Mass Spring Systems are known to be simple and easy to use, however they require strong physical basis to be considered suitable for accurate simulations. On the other

hand the medical environment constantly encourages research in the direction of real-time simulation of real, deformable bodies. The presented methods allow us to globally control the non-linearity and volume of a locally-focused Mass Spring System and improve its precision. Additionally, the mixed-element mesh implementation will facilitate accurate simulations with real-time performance on a minimal required number of elements.

Acknowldgements

For three years I have worked hard to obtain the PhD title. If it's been a success, it most definitely wasn't solely for my work alone. I cannot count the times I asked for help or advice and, sure enough, received more than I ever hoped for: friendship, understanding, support. As every scientist shall back my words if asked - we are merely people with ideas, but it is the bright insight, countless discussions and unique views presented by others that make our work possible.

On this note I would like to thank my irreplaceable team of supervisors. Guillaume for offering me help when I could see no further options, for not throwing me out of his room when I took a peak there for the thousandth time, for showing me how many things can be 'templated'. Florence, true master of all articles and bibliography, for sharing her enormous resources and broad knowledge with me. Stéphane, for the practical, 'hands on' aspect of our work, the one which drew me most to this topic. Jean-Francois for his stunning ideas, mathematical reasoning and unwavering certainty, all used to introduce us to the mechanical understanding of our work. Working with each of you was pure pleasure, even when it was hard work!

I would like to thank my lab-friends for all the time spent together. For the cakes, cookies and only a bit of coffee. I wish to see you all every time I come to visit! I couldn't foresee meeting people who became such an important part of my life: Abdou, Ewelina, Patrycja, each of you holds a special place in my heart. Amandine, thank you for being with me ever since we just had each other. Ania, Becia - for not closing the book on me, even when we saw each other only so often.

The agreement of the personal and professional aspects of life is of a great importance to me. I believe that nobody should be forced to give up one in favor of the other. I believe it so much so that the biggest achievement of the last three years of my life is not only the successful defense of my doctoral thesis, but also marrying the love of my life and becoming a mother. That said, I would like to thank Damian for his unabated support of my (even most crazy) ideas and for making my dreams come true. My parents for their incredible encouragement, which lasted even if you knew we'd see each other only twice a year. Thank you for the bond we have, the one no one with their right mind can understand.

I wish to thank everyone who I have met on my journey, for being part of it.

Contents

List of Figures

Dedicated to my Mother who believes in me no matter what…

Chapter 1

Introduction

Motivation - Soft Tissue Modeling for Biomedical Simulation

Nowadays the need for simulation, so modeling real-life events and phenomena, arises with development of many scientific branches. Being able to predict particular events or behaviors (like weather or interactions between bodies) under known circumstances is a valuable asset in treatment of many problems. It strongly affects the decision-making process and makes a huge impact on one's actions. Among many types of physical simulation we can distinguish simulation of gases, liquids, clothes, rigid and deformable bodies. The main interest of this thesis lies in simulation of deformable solids. Deformable objects like soft bodies and, in particular, soft tissues like internal organs are modeled to improve patient treatment and study nature of the human body.

Now, let us introduce different aspects of the physical simulation depending on the field-type point of view: biomedical simulation in computer graphics and biomechanics.

Biomechanical Simulation in Computer Graphics

As many authors like to remind, the pioneers in the field of *simulation* in computer graphics are Demetri Terzopoulos, John Platt and Kurt Fleischer, who in late 80's presented first deformable models [150, 149].

The application vary from offline to real-time simulations to model visually and physically accurate models of liquids, rigid and deformable bodies. As the field includes such a vast number of areas in which it can be used, we can choose and pick from various types of simulations including cloth animation, fluid simulation, rigid objects and soft tissue behavior under interactions with other objects. To study this field in depth it is useful to be equipped in the knowledge of the basic elasticity theory and continuum mechanics. The fields using these types of simulation the most are of course gaming and movie industry. Therefore the most important features from the computer graphics point of view are: time performance, visual accuracy and lightweight solutions.

There are several approaches to simulate realistic objects in real time, like learning algorithms or random function generators, but the one we focus on is a simplified physical-mathematical solution. This approach gained great popularity. Usually methods like Mass-Spring System or Position Based Dynamics are used to model even complex objects. They are said to be physically inaccurate, but produce real-time simulations using simplified physical laws, mathematical models and many approximations.

Biomedical Simulation in Biomechanics

Unlike computer graphics, the biomedical and especially biomechanical community values the accuracy of the simulation and its robustness. Therefore, to create complex systems for simulation, most of the researchers use Finite Element Model (FEM) and its variations. FEM is used to solve the equations responsible for body's mechanical behavior. These simulations may take very long time, but are known to be reliable when it comes to the solution. The biggest interest includes simulation of deformable bodies meaning mostly soft tissues, which can be used for medical training and virtual surgeries. An early survey on the difficulties in creating realistic models for medical simulation of soft tissues was written by Delingette in 1998 [45], to which we direct a curious reader.

The biomechanical simulation field involves several pioneers especially worth mentioning. First of all there are Carley Ward and Robert Thompson, who in 1975 introduced a detailed Finite Element model of a brain [158]. They based the model on fewer assumptions while keeping it detailed enough for future brain simulations and the study of brain dynamics. Then, there are Cover et al. who in their paper from 1993 [36] talked about gall-bladder model for surgical simulation and their methodology which addresses issues of graphical models for surgery simulation in general. Shortly after that, in 1995, Bro-Nielsen presented an active cubes approach used to model deformations of human tissue in response to bone movements [26]. We cannot forget also about the ever-active Stéphane Cotin, who in 1996 was among first to introduce us to the virtual surgery environment and application of tissue modeling in the biomedical context [35].

To find out about the mechanical properties of the modeled tissues, the physical behavior of the tissue needs to be studied in depth. Usually to obtain information about the simulated tissue it is necessary to perform numerous experiments on real tissues like human or animal organs. The tests include treatment in different types of loads: compression, traction, torsion, etc. Specially designed devices measure necessary parameters to obtain different curves (*e.g.* stress-strain), which depict the behavior of studied tissue. From those it is possible to derive constitutive laws [82, 124, 32, 94, 172, 126] which then can be used for example by Finite Element simulators.

Since 1967 Fung studied in depth the elastic behavior of soft tissues [62] and the mechanical properties of real bodies. The complexity of soft tissues includes many properties, which he studied by mechanical experiments [59]. These experiments have proven many biological tissues to manifest properties like incompressibility (or behavior very close to incompressible) [161], anisotropy [59], visco-elasticity and non-linearity [46, 61, 59]. From Fung we also know that these bodies exhibit non-linear behavior when they undergo large deformations [59].

According to the application, a good approximation of the soft tissues behavior is obtained by making some assumptions such as isotropy [100], elasticity [145, 29] and homogeneity [75].

Medical simulation from early 80's to this day

Since the possibility of simulation appeared on the horizon, the medical environment got also interested in it. More specifically, a solution for a gap in the training process of the health professionals was discovered. Since early beginning of medicine the study was first only theoretical, to then pass to the practical approach as an assistant or an intern and finally, to become self sufficient physician on an independent job. This learning approach involves major risk for potential patients, therefore this is where the medical simulation comes in view. It allows an accurate study of real bodied and what kinds of behavior one

can expect from different tissues when dealing with medical examination, recognition of abnormalities and even surgery. It provides a physician's understanding of the processes which take place in a body and the use of medical equipment and tools. Additionally, such a training will increase the safety of a patient and minimize the risk that comes with complex surgeries.

First mention of a simulation software were reported in Computer Gaming World magazine by Boosman [23], when the game *Surgeon* appeared on the market. The game simulated only one condition: aorta aneurysm. It allowed the user to cut through the tissues and perform a surgery including actions like removing the sick organ, replacing it with a healthy one and closing up a wound. The game's accuracy was judged as very realistic, especially at this level of computer development. Other softwares followed this predecessor and allowed smooth development in the field.

Next major breakthrough was founding of Society for Simulation in Healthcare in 2004, which consists of professionals who care about the quality of education, testing and research in general through medical simulation [1].

Finally, the simulation-based medical educational research reaches many institutes and universities with high fidelity software featuring haptic devices, physical examination models and virtual environments. Among them there are the well known products of Surgical Science or VirtaMed (Fig. 1.1) or medical simulation frameworks like SOFA [3]. The medical simulation proves to be overall less costly, allows to track the improvement of the trainee, it is more efficient in training and time performance and allows to avoid risk in the later patient treatment.

FIGURE 1.1: Left: EndoSim Flexible Endoscopic Simulator from Surgical Science, Inc. Model number ES01; right: VirtaMed ArthroS with shoulder model from VirtaMed AG.

Problem Statement and Aim of the Thesis

In the presented context soft tissue simulation became a huge field of interest in the biomedical environment as it provides reliable information about tissue behavior and allows to create complex models of human intestines involving many different organs at the same time. The challenge in fast soft tissue simulation is to faithfully represent the simulated tissues and their non-linear behavior at large deformations. Additionally, global behavior of real organs, like volume changes, is hard to control. There are many simulation methods, all of them with their advantages and limitations, which we present in the following chapter. The choice of a simulation method also poses many questions - the aimed simulation must be looked at from all the different angles: type of simulated material, stability, time performance, computational capabilities, level of detail and use of small or large time steps. The created framework could then be integrated as a part of larger simulation environment with use of haptic devices.

Our interest lies in a real-time biomedical simulation, where simplified deformation models are used. We are also interested in parallelization approaches on both CPU and GPU and possible pre-computations. Fast and accurate simulation shall then be used to simulate real organs for medical surgery simulation. The existing approaches present limited possibilities when it comes to type of simulated tissues and behaviors they represent. Usually there are three main issues we meet:

- use of simplified laws which do not succeed at modeling deformable bodies with enough accuracy for medical purposes;
- use of complex formulas which fail at real-time simulation;
- use of computation acceleration technique based on moving possibly biggest number of calculations to pre-computational phase of a simulation, which prevents use of topological modifications in an ongoing simulation.

Our work aims at creating a framework capable of performing simulations of deformable bodies like real organs in real time. The goal of this thesis concerns thus the implementation of a new biomedical model of soft tissues suitable for interactive simulations and allowing simulations of organ movements. For that purpose we decided to improve a locally-oriented Mass Spring System to accurately model real organs at large strains and control the volume of the simulated tissue. We have chosen Mass-Spring System as the simulation method because of its simplicity, ease of use and implement new solutions and fast performance. However, due to natural limitations of this modeling technique we have decided to introduce some improvement to reach our goal. The first task aimed at simulating small (linear) or large (non-linear) deformations without specifying which of them a body undergoes. Non-linearity is one of main characteristics of real bodies at large deformations. Since MSS is a locally-oriented technique with so called Cauchy limitation, our second task included controlling global feature: volume of a deformed tissue defined with compressibility constraint. The third task aims at expanding the variety of mesh types which our MSS is able to use. The new type of mesh involves mixed-elements: hexahedra, tetrahedra, prisms and pyramids.

Contributions

In this thesis we have investigated various methods which deal with the challenge of soft tissue simulation, especially in the context of medical application. In our work we focus on aspects like genericity, accuracy and physically-based inspiration. Our methods are simple to use and implement, same as the Mass-Spring System itself. They improve the MSS and broaden the possibilities to use it for fast, stable and controllable simulation.

The major contributions of this thesis are:

- **Non-linear force formulation for simulation of real organs inspired by a constitutive law.** This method aims at reproducing a real experiment used to formulate a bi-power constitutive law. The new formulation of force allows us to control the shape of a stress-time curve at different strain rates when deforming a visco-elastic body.
- **Physically-based formulation of stiffness for all-hexahedral mesh.** We present a physically-based derivation of the stiffness parameters for three types of springs in an all-hexahedral MSS mesh. The formulation is independent of the type of deformation the simulated body undergoes and it stabilizes MSS to avoid a natural wrinkle and buckling phenomena. The formulation considers elastic, homogeneous, isotropic tissues.
- **New volume correcting force based on the Poisson ratio.** This contribution allows us to control the volumetric changes in MSS depending on the compressibility rate parameter called Poisson's ratio (ν). It aims at dealing with natural MSS limitation which restricts the physically correct 3D MSS to such with $\nu = 1/4$. Additional force terms computed at every time step when a volume change occurs correct the resultant error at $\nu \neq 1/4$ and allow to use values of this parameter characteristic of natural materials: $\nu \in [0; 0.5[$.
- **Introduction to formulation of mixed-element mesh for Mass-Spring Systems.** We propose to formulate the physical and topological structure for a MSS mixed-element mesh. Such a structure consists of four basic 3D shapes: commonly used tetrahedra and hexahedra and joint-elements used to connect them - pyramids and prisms.

Organization of the Thesis

The thesis outline is the following:

- Chapter 2 presents the overview on the past research done in the field of physical simulation and modeling deformable bodies, also for medical purposes. We analyze a general simulation loop with existing time integration methods as well as the simulation techniques and how they are improved to successfully model behavior of elastic solids.
- Chapter 3 describes in detail Mass-Spring Model. It includes detailed information on simulation loop, MSS limitations and instabilities and types of mesh used for simulation. In further part we talk about different methods to obtain the physical MSS parameters and constraints used to control various MSS features. We discuss the non-linearity of MSS and possible topological modifications. Finally we present

our framework used for MSS simulations and the initial implementation of the simulation method.

- Chapter 4 introduces our novel approach used for simulation of real organs up to 100% strains. The simulation is as fast as the original MSS and is able to control the shape of the output stress-time curve at different strain rates. The method reproduces the shape of a visco-elastic body deformation thanks to minimal number of tissue parameters.
- In Chapter 5 we present an improvement of the MSS which provides two main features: a solution to avoid wrinkle and buckling instability as well as volume correcting forces. Such an approach provides us with a stable, physically-based MSS able to control the global volumetric changes of a simulated body based on a compressibility parameter. It successfully controls the volume of absolutely compressible to incompressible soft elastic, homogeneous, isotropic bodies.
- Chapter 6 proposes a mixed-element mesh structure and its parametrization for MSS. This type of a lattice would allow a faster simulation while providing detailed mesh with a minimal number of used elements.
- Finally, Chapter 7 concludes the thesis by summarizing the research results and discussing the future possibilities and directions of the next steps.

Collaboration and Publications

This work was supported by the LABEX PRIMES (ANR-11-LABX-0063) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR).

This thesis allowed a collaboration between Computer Science Laboratory for Image Processing and Information Systems (Laboratoire d'InfoRmatique en Image et Systèmes d'information - LIRIS) and The Biomechanics and Impact Mechanics Laboratory (Laboratoire de Biomécanique et Mécanique des Chocs - LBMC) of which representatives are Florence Zara, Guillaume Damiand (LIRIS) and Stéphane Nicolle (LBMC). The collaboration was also extended to Physic laboratory of ENS (École Normale Superieure) represented by Jean-François Palierne.

The topic of this thesis is 'Hybrid 3D Mass Spring System for Soft Tissue Simulation', which expands capabilities of the simple Mass Spring System to model complex behaviors of deformable bodies. This thesis introduces improvements of the simulation technique to deal with its natural limitations.

Our publications involve the following articles published in conference proceedings or submitted for review in journals:

- Karolina Golec, Matthieu Coquet, Florence Zara, Guillaume Damiand. 'Improvement of a Topological-Physical Model to manage different physical simulations', 23rd International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2015, Jun 2015, Plzen, Czech Republic. pp.25 – 34, WSCG Full papers proceedings;

- Karolina Golec, Florence Zara, Stéphane Nicolle, Jean-François Palierne, Guillaume Damiand. 'New Mass Spring System formulation to model the behavior of soft tissues', 22nd European Society of Biomechanics Congress (ESB 2016), Jul 2016, Lyon, France;
- Karolina Golec, Jean-François Palierne, Florence Zara, Stéphane Nicolle, Guillaume Damiand. 'Hybrid 3D Mass-Spring System to simulate isotropic materials', Work In Progress Session, VRIPHYS 2017: 13th Workshop on Virtual Reality Interaction and Physical Simulation 2017;
- Karolina Golec, Jean-François Palierne, Florence Zara, Stéphane Nicolle, Guillaume Damiand. 'Hybrid 3D Mass-Spring System for simulation of isotropic materials with any Poisson's ratio', under review in The Visual Computer International Journal of Computer Graphics.

Chapter 2

Simulation of Deformable Objects

In this chapter we introduce previously employed physically based models in the topic of soft tissue modeling for biomedical simulations. We describe the general context of soft tissue simulation in biomechanics, research in the direction of various techniques and the acquisition of tissue parameters. Then, we remind the basic notions of mechanics including continuum mechanics and commonly used time integration schemes. We finish the chapter by introducing three main simulation methods, their advantages and weaknesses.

2.1 Deformable Objects in Biomechanics

Soft bodies are known for several types of behavior which define them: they are incompressible, visco-elastic, their stress-strain relationship depends strongly on the type of tissue and conditions of the physical experiments [60]. The knowledge about mechanical behavior of human body is crucial in various fields like crash-testing, biomedicine or sport simulations. Identification of mechanical properties of different tissues allows us to know the stresses acting within them during an impact as well as compute the haptic forces used in virtual surgeries.

Deformable objects can also be defined in terms of linear and non-linear stress-strain relationship, meaning that their behavior is linear if the body undergoes a small deformation and strongly non-linear if the body is deformed above the linear limit. Soft tissues can be studied under different angles. An example of several relationships is depicted in Fig. 2.1, where the curves are plotted for a mono-axial tensile experiment of cartilage tissue [168].

2.1.1 Physical Parameters of Soft Tissues

Study of mechanical behavior of human organs is a crucial task in simulation because of the severity of a potential trauma [166, 117], meaning extensive bleeding and the fact, that they play the key role in the functioning of human body. The topic is frequently discussed in the context of surgical simulation as well as accidentology and automotive safety. Right after the brain tissue [108, 109, 125, 74, 154] the kidney and liver [98, 118, 139, 142, 120, 90] are reported most often in terms of their mechanical behavior, followed by the spleen in the same kind of identification [146, 138, 87].

The acquisition of the mechanical properties of abdominal organs is useful especially when applying or developing constitutive laws for FEM simulations. Several existing solutions for this type of organ simulation in automotive safety are described by Nicolle et al. [123] including, but not limited to: Ford model from Ford Motor Company [53], Humos model from the HUMOS European Consortium [155] or THUMS model from Toyota Motor Corporation and Toyota Central R&D [143].

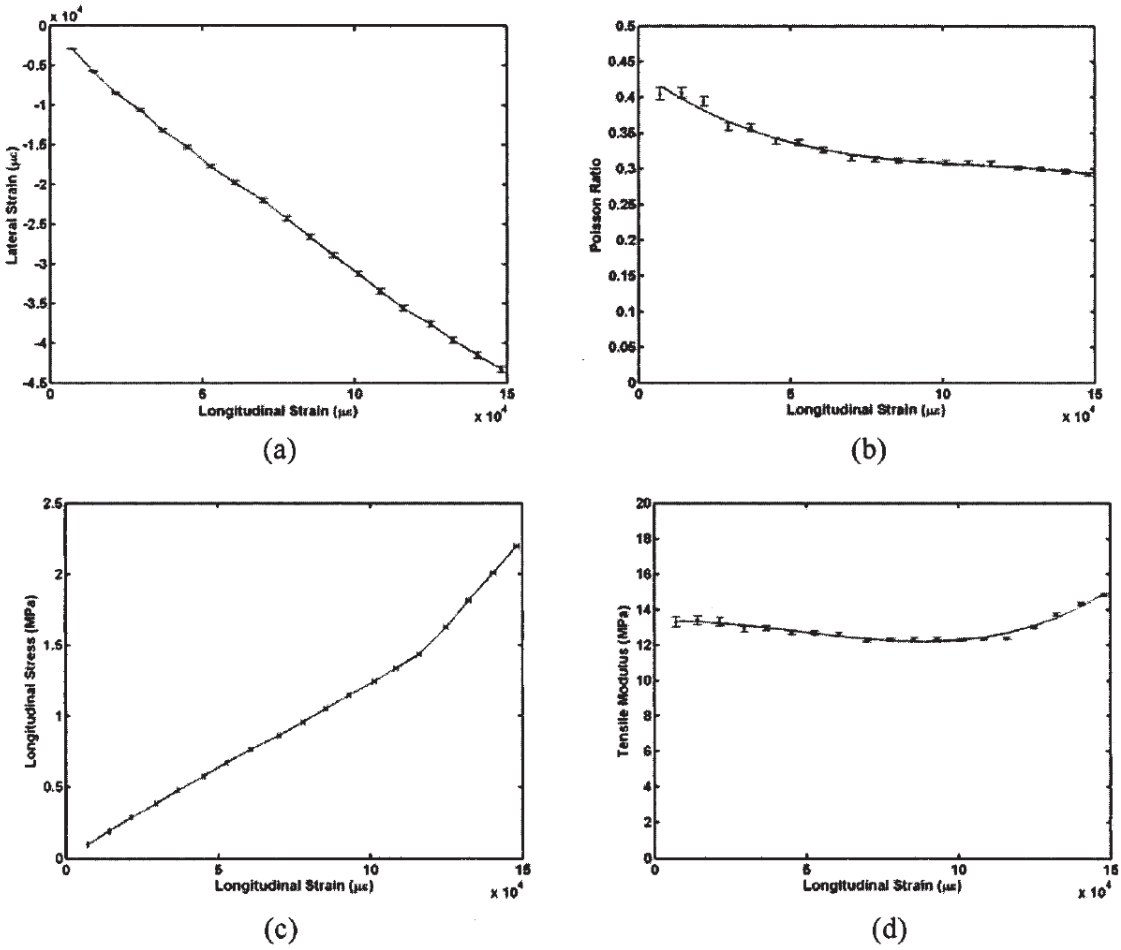


FIGURE 2.1: Relationship curves relative to longitudinal strain for (a) lateral strain, (b), Poisson's ratio, (c) longitudinal strain and (d) tensile modulus [168].

Additionally, in the context of medical simulation, we know that to be able to simulate the behavior of soft bodies accurately on haptic devices it is crucial to know the necessary level of accuracy of a simulation. Study of Batteau et al. shows that the perception of haptic feedback varies greatly, even among the specialists [11]. In their experiments they found out that the standard deviation of such a perception is within 8%. Moreover, Misra et al. have shown that the non-linearity of a tissue deformation highly exceeds the human perception threshold for force discrimination in some tissues and thus has to be taken into account in surgery simulation [111].

2.1.2 Application of Soft Tissue Modeling

The application of the soft tissue modeling can vary as well: we can distinguish models used purely for a full surgery simulation or surgery-guidance systems, which are used intraoperatively *i.e.* during an ongoing surgery. A survey on particular type of modeling: real time deformable models for surgery simulation, was written in 2005 by Meier et al. [106]. The authors gathered many simulation techniques available for modeling elastic objects in real time, dividing them, however, into two categories: interactive or realistic. Another survey, this time in the topic of computer-aided surgery, was written by Frisoli et al. [57]. The authors discuss two main aspects of this kind of simulation: the technological and realistic simulation challenges. We direct a curious reader to these articles for more details. Now let us review several contributions made in these areas.

Keeve et al. studied the behavior of MSS and FEM for **craniofacial surgery simulation** [86]. This application took into account the complexity of human tissue, which in this case was constructed of 3 layers with different properties, and the underlying bone structure, which plays the most important role in the postoperative result to be simulated. Both simulation methods turned out to allow a correct prediction of the impact of the applied changes, even though the MSS was known to be not as accurate as FEM, especially FEM in its expanded form including visco-elasticity and long-term relaxation of the tissue.

MSS-Based Surgery Simulation

A method created for **real-time micro-surgery simulation** of blood vessels was presented by Brown et al. [28]. The Mass-Spring System based method favors the visual rather than numerical accuracy and is focused on local deformations only, but this early application of MSS in virtual surgery was one of the first to take into account the properties of real tissue such as **visco-elasticity**, **non-linearity**, **non-homogeneity** and **anisotropy** instead of their respective simplifications.

An intraoperative update of the **surgery guidance** using ultrasound segmentation combined with MSS was presented by Dagon et al. [39]. Their simulation system was built of two elements: update of the positions of simulated liver tissue and its vessels; and a simple elastic deformation system used to model organ deformations, so the updated mesh. The idea was first developed on Matlab software and the accuracy was not taken into account at this point yet, since all the parameters of MSS were adjusted manually.

FEM-Based Methods

Niroomandi et al. put up with several challenges of simulation of deformable bodies, like **non-linear behavior** or **real time execution**, using Proper Orthogonal Decomposition (POD) technique [130]. Their experiments have shown good alignment with the complete model results. However, the method limits the simulation to very few degrees of freedom

and suffers from buckling phenomenon (we discuss the buckling instability in Ch. 3.2.2), which causes an error up to 30% in large deformations. Additionally the method requires a lot of pre-computations which may potentially cause difficulties in a simulation of topological modifications. Nevertheless, the authors propose several methods for simulation of deformable bodies including a Proper Generalized Decomposition (PGD) for simulation of **non-linear hyper-elastic solids** and **soft tissues** [37, 128] or reduced order modeling with X-FEM [129].

A deformable model for **virtual-reality** based simulations of Choi et al. was built on an idea of a deformation understood as a localized force transmittal process [31]. Their algorithm is thus based on a breadth-first search with an adjustable depth and uses simulated annealing process with linear static FEM reference data.

Other Techniques

Virtual surgery training system, which used a meshless technique and a method of finite spheres combined, was presented by De et al. [44]. The demonstrated real time simulation have assumed behavior of **linear elastic tissue** and aimed at the speed of the simulation rather than anything else.

As we have shown, in the medical simulation of deformable bodies there are various forms of application of the developed methods. The field of virtual surgery introduces many types of behavior into the surgery room with computer-aided procedures and virtual reality training programs. Since human tissues manifest very different types of behavior and physical properties, the simulation capabilities have to be adjustable and precise to meet the today's requirements for a good surgery simulation method.

2.2 Basic Notions of Mechanics

In this section we present the basic notions of mechanics and continuum mechanics to introduce the methods used by different simulation models in a dynamic simulation.

2.2.1 Simulation Loop

In this context we need to use the Newton's second law of motion, which writes:

$$\mathbf{a}\, m = \mathbf{f}(\mathbf{v}, \mathbf{x}, t), \tag{2.1}$$

where $\mathbf{a}$ is the acceleration, m the mass, $\mathbf{v}$ the velocity and $\mathbf{x}$ the position of an object at time t. $\mathbf{f}$ is a general function dependent on the physical model. It is defined by $\mathbf{v}$, $\mathbf{x}$ and t parameters. Then, an integration scheme of a choice is used to evaluate the velocities $\mathbf{v}$ using accelerations $\mathbf{a}$ and then calculate the resulting positions $\mathbf{x}$ from the previously obtained velocities. This is usually broken down into a general simulation loop, where the following components are integrated:

$$\begin{aligned} \mathbf{a} &= \frac{\sum \mathbf{f}(\mathbf{v}, \mathbf{x}, t)}{m}, \\ \mathbf{v} &= \int \mathbf{a}\, dt, \\ \mathbf{x} &= \int \mathbf{v}\, dt. \end{aligned} \tag{2.2}$$

This can be written down in a form of a system of first order differential equations. At the end, a chosen integration method needs to solve a system of ordinary differential equations (ODEs) of a form:

$$\begin{cases} \mathbf{v}_n = \mathbf{f}(\mathbf{v}_n, \mathbf{x}_n, t_n) \\ \mathbf{x}_n = x_0 \end{cases} \tag{2.3}$$

taking into account the positions x_n and velocities v_n at time t_n for unknown set of discrete values $n \in \{1, \ ..., \ N\}$, where n marks the consecutive time steps and N is the number of simulation steps.

2.2.2 Continuum Mechanics

Now, we present some basic notions about the continuum mechanics, which have to be covered to present the simulation methods.

From the Newtonian dynamics and Eq. (2.1) we can transform the second law of motion to a continuum mechanics manner, where the partial differential equation (PDE) of an *elastic, continuous* material is given by the Cauchy momentum equation:

$$\rho\ddot{\mathbf{x}} = \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} + \mathbf{F}_{ext}. \tag{2.4}$$

Here ρ is the body density and $\mathbf{F}_{ext}$ are the external forces acting on it. In this context the global forces $\mathbf{F}$ are usually computed as a derivative of the system's energy over a displacement vector $\mathbf{u}$:

$$\mathbf{F} = -\frac{\partial W}{\partial \mathbf{u}}. \tag{2.5}$$

Knowing that we use the linear algebra and calculus to describe deformations on continuous or discretized objects, we can describe a deformation Φ as a *displacement field* applied to an undeformed body $\Omega \in \mathbb{R}^3$. Then, when a part of the body at a location $\mathbf{X} \in \Omega$ is moved to a new location $\mathbf{x} \in \Omega'$ in deformed state Ω', we can define the *displacement vector field* as $\mathbf{u} = \mathbf{x} - \mathbf{X}$. To measure the deformation there exists a deformation tensor, which allows us to quantify it: Green Lagrange tensor. There are several choices as for how to compute the elastic infinitesimal Green-Lagrange strain tensor $\boldsymbol{\varepsilon}_{GL}$, which in the simplest (1D) case is a ratio of the change of the length divided by the initial length. The Hooke's law gives us the relationship between $\boldsymbol{\varepsilon}_{GL}$ and the stress $\boldsymbol{\sigma}$. It also gives us the the fourth-rank elasticity tensor $\mathbf{C}$, so that next we can choose the appropriate $\boldsymbol{\varepsilon}$:

$$\boldsymbol{\sigma} = \mathbf{C} : \boldsymbol{\varepsilon} \ ; \ \boldsymbol{\sigma}_{ij} = C_{ijkl} \, \boldsymbol{\varepsilon}_{kl}, \ \text{ for } i, j, k, l = x, y, z. \tag{2.6}$$

The Green-Lagrange strain tensor is then defined by:

$$\boldsymbol{\varepsilon}_{GL} = (\nabla \mathbf{X} + \nabla \mathbf{u})^{\mathrm{T}} (\nabla \mathbf{X} + \nabla \mathbf{u}). \tag{2.7}$$

The linearized Green-Lagrange strain tensor $\boldsymbol{\varepsilon}_L \in \mathbb{R}^{3\times3}$, which is valid only for small strains, is defined as follows:

$$\boldsymbol{\varepsilon}_L = \frac{1}{2} \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathrm{T}} \right). \tag{2.8}$$

Another solution is to use Green's non-linear strain tensor $\boldsymbol{\varepsilon}_N \in \mathbb{R}^{3\times3}$ defined as:

$$\varepsilon_N = \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T + (\nabla \mathbf{u})^T \nabla \mathbf{u}). \tag{2.9}$$

In both cases $\nabla \mathbf{u}$ is a 3×3 displacement gradient tensor (spatial variation of the displacement field), where the $x,\ y,\ z$ indices indicate the differentiation over the $u,\ v,\ w$ components of $\mathbf{u}$:

$$\nabla \mathbf{u} = \begin{bmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ w_x & w_y & w_z \end{bmatrix}. \tag{2.10}$$

Having the previously introduced displacement vector field $\mathbf{u}$, Lamé parameters ($\mu = \frac{E}{2(1+\nu)}$, $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$) and the linear Green - St. Venant strain tensor (Eq. (2.8)) it is possible to compute the linear elastic energy W_l for *homogeneous isotropic* materials, *i.e.* Hooke's law:

$$W_l = \frac{\lambda}{2}(Tr(\varepsilon_L))^2 + \mu\, Tr(\varepsilon_L^2). \tag{2.11}$$

Let us focus on the linearized tensor and its linear elastic energy from Eq. (2.11) first. It is can take the following form, modeled by the Hooke's law, which shows that the linear elastic energy of a deformable elastic object is a quadratic function of the displacement vector [135]:

$$W_l = \frac{\lambda}{2}(\boldsymbol{\nabla} \cdot \mathbf{u})^2 + \mu||\nabla \mathbf{u}||^2 - \frac{\mu}{2}||rot(\mathbf{u})||^2, \tag{2.12}$$

where $\boldsymbol{\nabla} \cdot \boldsymbol{\sigma}$ is the divergence of the stress tensor $\boldsymbol{\sigma}$. the stress tensor derives from the strain energy function given by Eq. (2.11) in the linear elasticity theory. The non-linear deformations will then be defined by non-linear energy W_{nl} which corresponds to the St. Venant Kirchhoff model - extension of the linear elastic model to non-linear deformations. W_{nl}, defined with the non-linear Green tensor (see Eq. (2.9)), becomes a polynomial of 4^{th} order:

$$\begin{aligned} W_{nl} =& \frac{\lambda}{2}(Tr(\varepsilon_N))^2 + \mu\, Tr(\varepsilon_N^2) \\ =& \frac{\lambda}{2}\left(\boldsymbol{\nabla} \cdot \mathbf{u} + \frac{1}{2}||\nabla \mathbf{u}||^2\right)^2 + \mu||\nabla \mathbf{u}||^2 - \frac{\mu}{2}||rot(\mathbf{u})||^2 \\ &+ \mu\left(Tr(\nabla \mathbf{u}^T(\nabla \mathbf{u}^T \nabla \mathbf{u}))\right) + \frac{\mu}{4}||\nabla \mathbf{u}^T \nabla \mathbf{u}||^2 \end{aligned} \tag{2.13}$$

2.2.3 Integration Schemes

Let us continue the detailed mechanical introduction by the most popular first order integration schemes, which are a base of a dynamic simulation and allow us to predict the behavior of a body at a particular time step. We also present one second order integration method called Verlet integration scheme. In real time simulation one needs to carefully choose the integration method to use. Since the simulations are largely dependent on the time step the most crucial criteria are *stability* and *simplicity*. The scheme has to allow use of large number of elements, fairly big time step and a fast solving method.

Explicit Euler Scheme

Explicit Euler method, called also Forward Euler, is the simplest of the presented methods. It is also straight-forward in computation. It computes the information about the next time step explicitly with the following formula:

$$\begin{cases} \mathbf{v}_{n+1} = \mathbf{v}_n + \Delta t\ \mathbf{f}(\mathbf{x}_n, t_n) \\ \mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t\ \mathbf{v}_n \end{cases} \tag{2.14}$$

where $\Delta t = t_{n+1} - t_n$. The method is said to be 'blindly' iterating into the future and not taking into account the changing derivatives, while its implicit sibling computes the output values as a part of the current step equation, therefore solving the system instead. The advantage of the Forward Euler scheme is that only the function $\mathbf{f}$ needs to be evaluated at a time step.

Semi-implicit Euler Scheme

This method has many names including symplectic Euler, semi-implicit, semi-explicit Euler as the most common ones. As a symplectic integrator it gives better results than standard Euler method. The differential equations in this scheme take the following form:

$$\begin{cases} \mathbf{v}_{n+1} = \mathbf{v}_n + \Delta t\ \mathbf{f}(\mathbf{x}_n, t_n) \\ \mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t\ \mathbf{v}_{n+1} \end{cases} \tag{2.15}$$

The first noticeable difference between this and the explicit Euler is in the $\mathbf{x}_{n+1}$ equation, which here takes $\mathbf{v}_{n+1}$ in parameter, unlike the forward scheme, which simply uses $\mathbf{v}_n$. Another difference lays in the integration method. It turns out that the symplectic integrator almost conserves the energy of the system, which for the same case solved using standard Euler - increases steadily. One can easily draw a conclusion, that the accuracy of a system which conserves the energy is far better. However, both of those schemes need to use small time steps for stability, even though the computations at each time step are not complex.

Implicit Euler Scheme

As the name indicates, the implicit Euler scheme (called also backward Euler) involves solving the implicitly given unknowns. This means that the system of equations, now defined by $\Delta\mathbf{x} = \mathbf{x}_{n+1} - \mathbf{x}_n$ and $\Delta\mathbf{v} = \mathbf{v}_{n+1} - \mathbf{v}_n$ becomes:

$$\begin{cases} \Delta\mathbf{v} = \Delta t\ \mathbf{M}^{-1}\ \mathbf{f}(\mathbf{v}_{n+1}, \mathbf{x}_{n+1}, t_n) \\ \Delta\mathbf{x} = \Delta t\ \mathbf{v}_{n+1} \end{cases} \tag{2.16}$$

where $\mathbf{M}^{-1}$ is the global mass matrix and function $\mathbf{f}$ usually defined with Taylor expansion [6] as:

$$\mathbf{f}(\mathbf{v}_{n+1}, \mathbf{x}_{n+1}, t_n) = (\mathbf{f}_n + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Delta\mathbf{x} + \frac{\partial \mathbf{f}}{\partial \mathbf{v}}\Delta\mathbf{v}). \tag{2.17}$$

In this equation we can already note important components: $\frac{\partial \mathbf{F}}{\partial \mathbf{x}}$ is the so called Jacobian in terms of positions and $\frac{\partial \mathbf{F}}{\partial \mathbf{v}}$ is the Jacobian in terms of velocities. Now, knowing that $\Delta\mathbf{x} = \Delta t(\mathbf{v}_n + \Delta\mathbf{v})$ and $\mathbf{M}^{-1}$ is a diagonal matrix, we can transform equation (2.17) into:

$$(\mathbf{M} - \Delta t\frac{\partial \mathbf{f}}{\partial \mathbf{v}} - \Delta t^2\frac{\partial \mathbf{f}}{\partial \mathbf{x}})\Delta\mathbf{v} = \Delta t(\mathbf{f}_n + \Delta t\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathbf{v}_n). \tag{2.18}$$

This system can be now evaluated as a matrix - vector system $\mathbf{A}\Delta\mathbf{v} = b$, with $\Delta\mathbf{v}$ as the vector of unknown velocity values. Solving it allows simple computation of velocities and subsequent positions of the point masses. Such a system can be solved using different factorization methods, like the most common LU factorization, or approximation one - as conjugate gradient [6].

Verlet Integration

Verlet integration is a second order integration method, which uses central difference approximation to the second derivative:

$$\frac{\Delta^2\mathbf{x}_n}{\Delta t^2} = \frac{\frac{\mathbf{x}_{n+1}-\mathbf{x}_n}{\Delta t} - \frac{\mathbf{x}_n-\mathbf{x}_{n-1}}{\Delta t}}{\Delta t} = \frac{\mathbf{x}_{n+1} - 2\mathbf{x}_n + \mathbf{x}_{n-1}}{\Delta t^2} = \mathbf{a}_n = \mathbf{A}(\mathbf{x}_n). \tag{2.19}$$

Such a representation allows us to compute the system of equations without velocities of the following form:

$$\begin{cases} \mathbf{x}_{n+1} = 2\mathbf{x}_n - \mathbf{x}_{n-1} + \mathbf{a}_n\Delta t^2 \\ \mathbf{a}_n = \mathbf{A}(\mathbf{x}_n). \end{cases} \tag{2.20}$$

However we can approximate the velocities $\mathbf{v}$ at step $n + \frac{1}{2}$, so in the middle of time-step interval. To do it we need to establish the velocities at time-steps $n + 1$ and $n - 1$ using Taylor expansion:

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \Delta t\ \mathbf{a}_n + \frac{\Delta t^2}{2}\mathbf{x}_n^{(3)} + \frac{\Delta t^3}{6}\mathbf{x}_n^{(4)} + \frac{\Delta t^4}{24}\mathbf{x}_n^{(5)} + O(\Delta t^5), \tag{2.21}$$

$$\mathbf{v}_{n-1} = \mathbf{v}_n - \Delta t\ \mathbf{a}_n + \frac{\Delta t^2}{2}\mathbf{x}_n^{(3)} - \frac{\Delta t^3}{6}\mathbf{x}_n^{(4)} + \frac{\Delta t^4}{24}\mathbf{x}_n^{(5)} + O(\Delta t^5). \tag{2.22}$$

Then, the difference of these terms equals:

$$\mathbf{v}_{n+1} - \mathbf{v}_{n-1} = 2\Delta t\ \mathbf{a}_n + \frac{\Delta t^3}{3}\mathbf{x}_n^4 + O(\Delta t^5), \tag{2.23}$$

which after derivation gives us the following formula for $\mathbf{v}_{n+1}$:

$$\mathbf{v}_{n+1} = \mathbf{v}_{n-1} + 2\Delta t\ \mathbf{a}_n + O(\Delta t^3). \tag{2.24}$$

Thanks to this computation we can directly find the initially searched approximation of velocity:

$$\mathbf{v}_{n+\frac{1}{2}} = \mathbf{v}_{n-\frac{1}{2}} + \Delta t\ \mathbf{a}_n. \tag{2.25}$$

Using the computation of velocities we can easily find the positions:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t\ \mathbf{v}_{n+\frac{1}{2}}. \tag{2.26}$$

It corresponds to the following:

$$\mathbf{v}_{n-\frac{1}{2}} = (\mathbf{x}_n - \mathbf{x}_{n-1})/\Delta t. \tag{2.27}$$

Shortly on Stability

Evaluation of the stability of every integration method facilitates the choice of appropriate scheme for particular simulation purpose. There exist several methods to compute the stability of a numerical integration scheme. In case of the first order schemes we can study a linear test equation

$$\dot{\mathbf{x}}(t) = \Gamma \mathbf{x}(t), \tag{2.28}$$

where $\Gamma \in \mathbb{C}$ is a parameter mimicking the eigenvalues of linear systems of differential equations, we test the stability by verifying if

$$real(\Gamma) \leq 0. \tag{2.29}$$

We know that the exact solution of Eq. 2.28 decays exponentially with time [68], meaning:

$$lim_{t\to\infty}\ \mathbf{x}(t) = 0. \tag{2.30}$$

Now, the verification is based on the same evaluation of a numerical solution $\mathbf{u}_n$, so when

$$lim_{n\to\infty}\ \mathbf{u}_n = 0. \tag{2.31}$$

Explicit Euler. We can evaluate the forward Euler scheme with the linear test equation:

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta t\ \Gamma\ \mathbf{u}_n \tag{2.32}$$

which gives us

$$\mathbf{u}_{n+1} = (1 + \Delta t\ \Gamma)^{n+1}\ \mathbf{u}_0. \tag{2.33}$$

In this case the solution is stable if $|1 + \Delta t\ \Gamma| \leq 1$.

Implicit Euler. Now, let us define the linear test equation in the backward Euler scheme as:

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta t\ \Gamma\ \mathbf{u}_{n+1} \tag{2.34}$$

which yields

$$u_{n+1} = \left(\frac{1}{1 - \Delta t\ \Gamma}\right)^{n+1} u_0. \tag{2.35}$$

From here we know that the solution is stable if $|1 - \Delta t\ \Gamma| \geq 1$.

The stability of backward and forward Euler schemes can be then depicted as shown on the well known charts in Fig. 2.2, where the stability regions are the colored ones.

Semi-implicit Euler. The stability of the semi-implicit Euler method can be evaluated in slightly more complicated way. The in-depth look at the stability of this integration scheme can be taken in the article of Niiranen [127], where the author comes with a conclusion, that the stability region of this scheme has it's center at $(2, 0)$ and is of a radius $2\sqrt{2}$, as shown in Fig. 2.3.

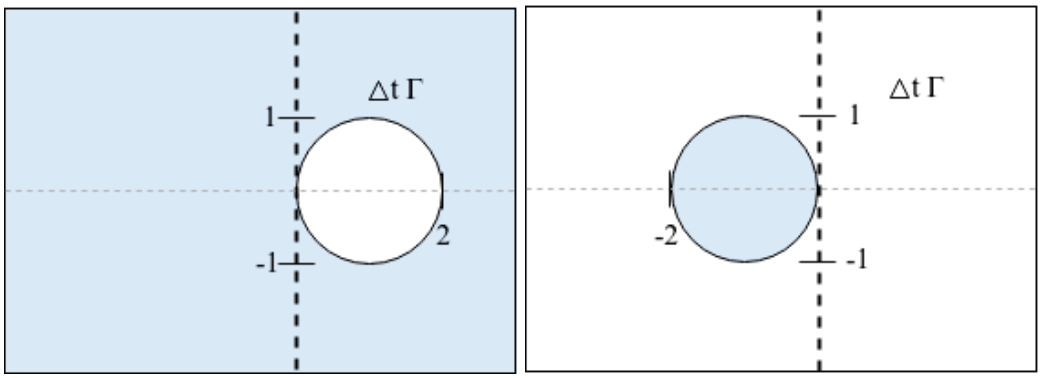


FIGURE 2.2: The stability regions for backward (left) and forward (right) Euler schemes; the colored regions indicate the stability.

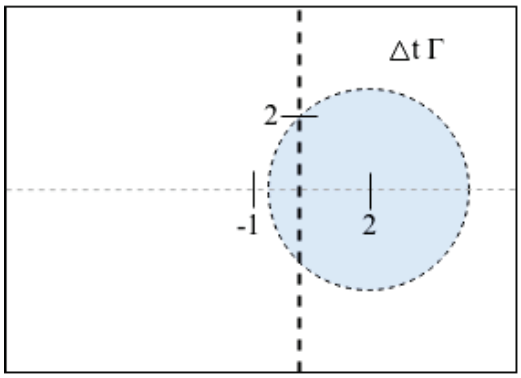


FIGURE 2.3: The stability region for semi-implicit Euler scheme; the colored region indicates the stability.

Verlet Integration Scheme. While in the forward Euler integration scheme a velocity is calculated by dividing it by the time step in consecutive steps (which causes a change of position, but also increases the error at every time step), the Verlet approach uses different method. In fact the Verlet integration assumes the time step to be constant and computes the velocity as the change in position by multiplying by the Δt. Thanks to this calculation difference the Verlet scheme is much more precise than the explicit Euler method.

Conclusions

There exist numerous more integration methods among which we have presented the ones which are used most frequently. In the literature there are available some improvements to the integration schemes to ameliorate the quality of simulation of deformable tissues [147, 162], also based on parallelization techniques [89]. The differences between the presented time integration methods focus on the stability and the size of the time step used for simulation. In general we can sum up the implicit integration scheme as unconditionally stable, which allows the use of time step of large size [6, 10], and the explicit as one suitable for use of small time steps only. The number of iterations is linearly dependent on the size of the time step, which has great influence on the calculation time. The semi-implicit time integration has a much better stability than its explicit counterpart and allows usage of small time steps.

2.3 Existing Simulation Methods

In the previous section we have covered the basis of mechanics and described the simulation loop used to calculate the movement of an object with a carefully chosen time step. Such a simulation loop is based on the computation of forces and depends on a physical model chosen for a simulation. We also presented several commonly used integration schemes for a dynamic simulation. Now we can continue by presentation of the most known simulation techniques, which use the notions presented before. Note that to choose the appropriate simulation method one needs to look from the point of view of the simulated body and the chosen conditions.

Simulation of deformable bodies is a broad field which covers topics from vector calculus, through topology and constraints acquisition until types of modeled behaviors. A great survey on the different types of models was written by Nealen et al. in 2006 [121] and earlier, in 1997 also by Gibson and Mirtich [65]. As the authors of the first survey [121] pointed out, the following subdivision of the methods can be made:

1. **Lagrangian Particle Methods** - this field can be divided into two parts: mesh based and mesh free methods. The first one is represented by Mass-Spring Systems and other continuum mechanics based methods, while the second one describes methods like Smoothed Particle Hydrodynamics (SPH) and other loosely coupled particle systems. In these methods, the models consist of particles with varying locations.

2. **Eulerian particle methods** - used mostly for gases, liquids and melting objects, with a statistical approach to solve the disperse phase. The particles are computed as a density in particular volume and treated as stationary points.

In this section we focus on three most commonly used simulation methods: Finite Element Method, Position Based Dynamics and Meshless Deformations.

2.3.1 Finite Element Method

The Finite Element Method (FEM), also referred to as Finite Element Analysis (FEA), described in detail in a book of K.J. Bathe [9], is the most popular choice for simulation in the exact sciences and medical environment. It computes approximations of real solutions to partial differential equations (PDE's), called basis functions. In FEM simulation we view simulated bodies as continuous volumes discretized into a sum of basis functions and solve the PDE's, which gives outstanding precision in the results. Thanks to this, FEM is used to simulate not only deformable bodies, but also airflow, heat transfer, magnetic flux, fluids, turbulences and many others. In general, the FEM problem is usually broke down into set of equations describing the properties of a body, boundary/initial conditions and particular body properties, such as elasticity, viscosity, etc. Then the system can be solved using the principle of minimization of the energy to obtain a static solution in an equilibrium position. The other possibility is to use dynamic simulation loop with one of the integration schemes presented in Ch. 2.2.

Application

One of good examples of application of the Finite Element Method is SOFA framework [3], where FEM is used for a dynamic medical simulation of various deformable objects including soft tissues.

As we already mentioned in Ch. 1, it was Terzopoulos et al. who first introduced simulation of elastic deformable models [150]. The FEM discretization method was first used in their paper to obtain set of partial differential equations (PDE's) of motion to be solved further on.

The usage of FEM to simulate deformable bodies was perfected by the authors over the years. In 1996 FEM was used by Bro-Nielsen and Cotin [27] in their 3D real-time simulation of deformable solids. They discussed three methods of solving the problem of real-time simulation with topological modifications using FEM. The authors focused on the speed of simulation without regard to the pre-calculation time or the accuracy of the simulation though.

In the same year, Cotin et al. presented virtual environment for surgery simulation, which used FEM as the basis [35]. The model was based on the elasticity theory and computed force feedback within the sample in real-time, however it was not simulating dynamical behavior.

Even more practical use of FEM for medical purposes was presented by Azar et al. in 2001 [5]. The FEM was used to simulate non-linear material properties of breast tissue to detect the position of a tumor after compression. Thanks to the predictions calculated by the method it was possible to detect the position of the tumor with 5 mm accuracy and use it for more accurate insertion of a needle in the actual procedure.

Simplifications and Improvements for Fast Simulation. In 1999 Berkley et al. presented one of the first simplified FEM methods designated for real-time simulation of anatomical structures [17]. The method was based on the preprocessing and optimization of the FEM mathematics and yielded only the essential simulation information.

In 2001 Wu et al. presented an adaptive non-linear FEM for simulation of deformable bodies improved by a global diagonal mass matrix for faster computations [163]. Classical non-linear FEM methods are known to be slow and the fast FEM simulations are based on approximations and lack standard FEM's accuracy. Wu et al. decided to use an improved non-linear method, which allows real-time computations. It used adaptive meshing to allow faster computations and keep the desired level of detail in the same time.

Many more authors attempted solving issues of real-time performance and accuracy in FEM to allow fast and precise simulation of deformable bodies and especially soft tissues. One can read more about mixed solutions with use of Finite Element and Finite Volume [43], FEM for large elasto-plastic deformations [77] and dynamic deformable simulation using FEM and modal analysis [8]. As we discuss later on in this chapter, the FEM method is most eagerly used for medical simulation of soft tissues and surgery planning, but still leaving room for many improvements.

FEM Derivation

Considering the use of FEM in the field of deformable body simulation, we have to know that the whole domain is divided into sub-domains (finite elements), for which the elementary problem is solved. The approximation of the solution is computed for the given mesh nodes and interpolated to the whole domain. Then after solving the problem for each sub-domain and gathering the solutions one can get the global solution itself. We already know, that the PDE of an elastic material is defined by Eq. (2.4). Next step is to define the spatially continuous function $\mathbf{x}(t) \in \Omega$ as a set of discrete locations $\tilde{\mathbf{x}}_i(t)$ (with $i \in \{1, ..., N\}$ and N being the number of nodes):

$$\tilde{\mathbf{x}}(t) = \sum_i \mathbf{x}_i(t)\mathbf{b}_i. \tag{2.36}$$

$\mathbf{b}_i$ are fixed binary nodal basis functions, or shape functions, meaning that they are equal to 1 at node i and 0 at other nodes. The general formula for a dynamic deformation of a mesh can be described as:

$$\mathbf{M}\,\ddot{\mathbf{x}}(t) + \mathbf{D}\,\dot{\mathbf{x}}(t) + \mathbf{K}\,\mathbf{x}(t) = \mathbf{F}_{ext}, \tag{2.37}$$

where the **M**, **D** and **K** are respectively global mass, damping and stiffness matrices.

Extended Finite Element Method

One of the most important FEM improvements is called XFEM. The XFEM, as presented for the first time by Belytschko et al. [13], is an eXtended Finite Element Method used for stable *interactive* simulation of deformable tissues. Unlike the classical FEM, XFEM method allows us to model cracks in the mesh without additional need for remeshing (or only with the minimal remeshing procedure) or re-computations connected with the boundary conditions. For the purpose of correct cracks modeling the so called enrichment functions are introduced into the XFEM system. Considering a crack, the neighboring j nodes of the crack are 'enriched', therefore the basic Eq. (2.36) is extended to the form:

$$\tilde{\mathbf{x}}(t) = \sum_i \mathbf{x}_i(t)\mathbf{b}_i + \sum_j \mathbf{b}_j\Psi_j\mathbf{a}_j, \tag{2.38}$$

where Φ is the enrichment function and **a** is the set of nodal degrees of freedom and $i, j \in \{1, ..., N\}$.

Even though the original purpose of XFEM was to be used on cracks and fractures on rigid surfaces [113, 148], it was lately adapted to model cutting in deformable objects [81, 97, 80, 85]. However the modeling of very curved surfaces requires use of a refined mesh [2].

Co-rotational Finite Element Method

Geometrically non-linear structures can be currently modeled with use of three Lagrangian kinematic descriptions: (i) Total Lagrangian, (ii) Updated Lagrangian and (iii) Co-rotational. The latter one is broadly used in medical simulators these days, therefore we shall take a closer look at it.

The *a priori* assumption of the co-rotational FEM is that *displacements and rotations may be arbitrarily large, but deformations must be small*. As the description follows the body as it moves, we can decompose the motion-tracking into two components - the base and the co-rotated configuration. While the base configuration serves as the origin of the body displacements, the purpose of the co-rotational one is to track the rigid body motion, where the associate coordinate system follows the body like a ghost. Then, all the deformations can be measured with respect to the co-rotated body configuration. In an element's local rotated frame the calculation of the forces applied by a deformed element to its sampling point is done using the following formula:

$$\mathbf{f} = \mathbf{R}^T\mathbf{K}(\mathbf{R}\mathbf{x} - \mathbf{x}_0), \tag{2.39}$$

where **K** and **R** are the global stiffness and rotation matrices respectively, **x** and $\mathbf{x}_0$ are the current and initial positions of the point.

The co-rotational FEM was first introduced by Wempner [159] and Belytschko and Glaum [14], as presented in the article of Nour-Omid and Rankin from 1991 [131]. The main idea is to compute the matrix **R**. Since the initial assumption of the co-rotational method poses a great limitation, the idea of multiple frames has been proposed by Veubeke [42]. Several other methods include use of eigenvectors and eigenvalues [114, 78, 54].

Admittedly, this method has significant superiority over the classical FEM, thanks to which it can easily be used in biomedical simulators. The CR layer works like a filter to model the deformations efficiently. However the main assumption limits the use of co-rotational FEM to only small deformations (assuming them to be smaller than 1%) and it does not involve interconnecting deformable and rigid elements.

Mass-Tensor Model

Another way of discretizing the equations of continuum mechanics is called Mass-Tensor (MT) or Tensor-Mass model. In Computer Graphics it was introduced by Cotin et al. [34, 33] and Picinbono et al. [135]. The method is based on another technique of discretization of the continuum mechanics equations.

The MT model is defined in two forms: small deformations (under 10%) are defined with the linearized Green-Lagrange tensor and the corresponding linear energy (see Eq. (2.12)) and big deformations ($> 10\%$) with Green's non-linear strain tensor used to compute non-linear energy (Eq. (2.13)). Next step is to compute the forces (using the aforementioned energies expressed with Finite Element Method) with Eq. (2.5). Then, for each 3D mesh element every force component has to be split into two parts. Thus, the force $\mathbf{F}_p^t$ of a particle p of an element $t \in N_p$ (where N_p is the number of volumes) of a mesh is a sum of the forces produced by the movement of the particle $\mathbf{u}_p$ and the forces coming from movements of its neighbors $\mathbf{u}_j$ ($j \neq p$). Therefore, for a single force component corresponding to a linear elastic case $2\sum_j \mathcal{B}_{pj}\mathbf{u}_j$ we have the following MT force:

$$\mathbf{F}_p^t = \mathcal{B}_{pp}^t \mathbf{u}_p + \sum_{j \neq p} \mathcal{B}_{pj}^t \mathbf{u}_j, \tag{2.40}$$

where $\mathcal{B}^t$ is a (3×3) local symmetric matrix of a linear elastic model dependent on the basic mesh element used, otherwise called local stiffness tensor. Now, to create the global tensors we need simply to sum up all the local tensors on the existing vertices and edges:

$$\mathcal{B}_{pp} = \sum_{t \in N_p} \mathcal{B}_{pp}^t, \tag{2.41}$$

$$\mathcal{B}_{pj} = \sum_{t \in N_{pj}} \mathcal{B}_{pj}^t, \tag{2.42}$$

where N_{pj} is the number of edges in the mesh. The other force components derived from the FEM formulation are responsible for the stiffness tensors of vertices, edges, 2D and 3D elements. The forces computed this way are used to update the positions of vertices from Eq. (2.37).

The MT model was used in real-tissue modeling like the methods presented by [142], and more recently by [93].

The most important limitation of the linear model is that it is not rotationally invariant. That can cause further issues with the reference frame when global rotations are applied and even bigger issues with the change of volume when only local rotations take place. MT is known to be not as slow as FEM itself thanks to the precomputed stress tensors when the mesh file is loaded, but it is also not as fast as the particle-based methods because of its computational complexity. Additionally, if the tensor functions are not constant, the computation time might increase even more.

2.3.2 Position Based Dynamics

The Position Based Dynamics (PBD) method of Matthias Müller was described first in his article [115] as a no-name approach and then introduced under its current name two years later [116]. In the field of rigid and deformable body simulation, Bender et al. published a state of the art report describing the large part of PBD methods of simulation [15]. Different constraint functions are used to simulate various types of materials and particular behaviors in PBD. Müller et al. [115] used rigid shape-matching constraint functions for rigid body simulation, which then were improved by Macklin et al. [104], coupled with an improvement of the simulation loop and stabilization step.

For a discretized object consisting of set of particles, the PBD, as the name already indicates, works directly with positions of those particles instead of their forces. Additionally the method is straight-forward and easy to use, especially in the gaming context. In the first attempts of introducing the PBD, Verlet integration scheme was willingly used by many authors, but only Müller introduced and elaborated on the whole idea.

Due to the similarities of Position-Based Dynamics approach to Mass Spring System (described in Ch. 3) as particle-based methods we want to elaborate on the PBD simulation method to point out and clarify differences between them.

General Algorithm

The most important role in Müller's PBD play the so called constraint functions $\mathcal{C}_j : \mathbb{R}^{3M} \rightarrow \mathbb{R}$ (not to be confused with $\mathbf{C}, C_{ijkl}$, the fourth-rank elasticity tensor), where M is the number of constraints in the system and $j \in \{1, ..., M\}$. The first type of the constraint function is called *equality*. As the name suggests it is satisfied if equality $\mathcal{C}_j(\mathbf{x}_1, ..., \mathbf{x}_N) = 0$, where N is the number of particles, is satisfied as well. The second type, called *inequality* is satisfied if $\mathcal{C}_j(\mathbf{x}_1, ..., \mathbf{x}_N) \geq 0$. The general algorithm of PBD consists of the following steps:

- initialization of the particle's variables (positions $\mathbf{x}$, velocities $\mathbf{v}$ and masses m);
- initialization/estimation of the new location for each particle (using explicit Euler integration scheme) according to the forces;
- iterative solving of the system by projecting the constraints $\mathcal{C}_j$ over the estimated positions (the projection is done using Gauss-Seidel method);
- moving the vertices to computed and corrected positions, updating the velocities.

This algorithm is indeed also based on the Verlet approach, but takes also from the modification of the current positions instead of storing the velocities as the difference between the positions from the current and previous step. Additionally it is interesting to note, that the iterative part of the algorithm makes it hard to define PBD as implicit or explicit. As the authors point out, the method will behave more like an explicit scheme

if only one iteration is used or more like an implicit one if the number of iterations is increased. The stability of this method is independent of the time step - the system is unconditionally stable. The only instability comes from the Newton-Raphson solver, which again is dependent on the shape of the constraint functions.

Stabilization Step

An improvement to the simulation loop was proposed by Macklin et al. [104]. Not only did they propose a parallelization of the algorithm, but they also introduced a stabilization step, which purpose is to fix the wrong positions of particles with contact constraints. Let us consider the example presented in Fig. 2.4. The presented particle of a wrong position at time $t = 0$ and initial velocity $\mathbf{v} = 0$ is fixed by their stabilization step before solving the system for the constraint functions. In the classical PBD this position would be fixed at step $t = 1$, however it would result in a velocity $\mathbf{v} \neq 0$, which then would propagate to the next step, as shown in the figure.

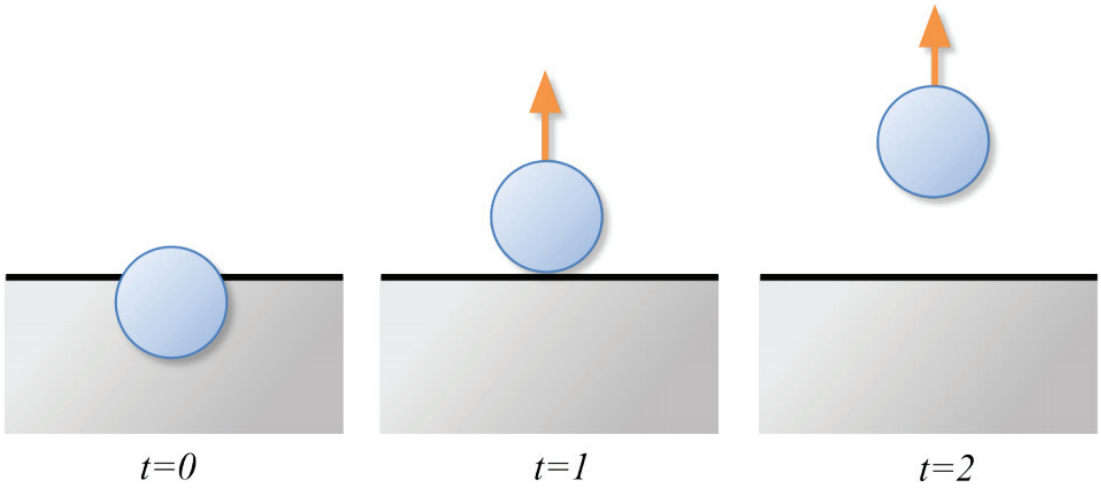


FIGURE 2.4: From left to right: a non-moving particle at time $t = 0$ being at a wrong position, which would be fixed at time $t = 1$, but with a resulting velocity propagating to the next step $t = 2$ [104].

Projection of the Constraint Functions - the Solver

The general treatment of the constraint functions is to find the corrections $\Delta\mathbf{p}$ to the predicted positions $\mathbf{p}$ of all the nodes affected by the constraint function C. This method is a well known non-linear Gauss-Seidel solver. Therefore, knowing the gradient $\nabla\mathcal{C}_\mathbf{p}$ and having the correction $C(\mathbf{p}+\Delta\mathbf{p}) = 0$, the solution can be derived from an approximation:

$$\mathcal{C}(\mathbf{p}+\Delta\mathbf{p}) \approx \mathcal{C}(\mathbf{p}) + \nabla_\mathbf{p}\mathcal{C}(\mathbf{p}) \cdot \Delta\mathbf{p} = 0, \tag{2.43}$$

which after solving it for a scalar multiplier a, such that

$$\mathbf{p} = a\nabla_\mathbf{p}\mathcal{C}(\mathbf{p}) \cdot \Delta\mathbf{p}, \tag{2.44}$$

gives the solution for $\Delta\mathbf{p}$ in agreement with an Newton-Raphson step:

$$\Delta\mathbf{p} = -\frac{\mathcal{C}(\mathbf{p})}{|\nabla_\mathbf{p}\mathcal{C}(\mathbf{p})|^2}\,\nabla_\mathbf{p}\mathcal{C}(\mathbf{p}). \tag{2.45}$$

This formula can be translated into a single position correction, with a scaling factor s and a weight $w_i = 1/m_i$ for a particle i:

$$\Delta \mathbf{p}_i = -s\ w_i\ \nabla_{\mathbf{p}_i} \mathcal{C}(\mathbf{p}). \tag{2.46}$$

The scaling factor s also involves a weight w, which is responsible for handling different masses in a non-homogeneous body:

$$s = \frac{\mathcal{C}(\mathbf{p})}{\sum_j w_j\ |\nabla_{p_j} \mathcal{C}(\mathbf{p})|^2}. \tag{2.47}$$

In case of a homogeneous body though, the weight would be set to 1 and not have any effect on the correction $\Delta\mathbf{p}$, however in the case of constrained particles, the masses are set to infinity, yielding $w = 0$ to not allow any movement.

Stiffness and Damping

The stiffness is a topic in PBD, which is not taken often into consideration. Since the most important in PBD is the *visual* accuracy, the incorporated stiffness is usually a value $k \in [0; 1]$. Then it is only the matter of multiplying the corrections $\Delta\mathbf{p}$ by k, which yields an error of $\Delta\mathbf{p}(1-k)^n$, where n is the number of solver iterations [116]. Usually, to get a linear relationship with a considerably smaller error, the correction is multiplied by a k-dependent term: $k' = 1 - (1-k)^{1/n}$. This method gives changes the error to $\Delta\mathbf{p}(1-k')^n = \Delta\mathbf{p}(1-k)$. It is interesting to note, that the resulting stiffness of the material will still be dependent on the time step, so the system will become arbitrarily stiff as the number of iteration increases or the time step decreases.

Similarly to the idea of Jeřábková and her XFEM, Macklin et al. [103] introduced the idea of XPBD - eXtended Position-Based Dynamics algorithm. The authors based their method on the Newton's equations of motion with an implicit time discretization. Then, instead of using the general Eq. (2.46), their update of the positions is given directly by computing:

$$\Delta\mathbf{p} = w_i \nabla \mathcal{C}(\mathbf{p}) \Delta a. \tag{2.48}$$

To make a connection of the new stiffnesses the authors used Gauss-Seidel update, evaluating the multiplier for stiffness:

$$\Delta a_j = -\frac{\mathcal{C}_j(\mathbf{p}_i) - \tilde{\alpha}_j a_{ij}}{\nabla \mathcal{C}_j\ w_j\ \nabla \mathcal{C}_j^T + \tilde{\alpha}_j}, \tag{2.49}$$

where $\tilde{\alpha} = \alpha/\Delta t^2$ and α is a block diagonal compliance matrix corresponding to inverse stiffness.

The *damping* in PBD however is dealt with better in the basic model already. First of all it is possible to use any damping model available in the literature [121]. However, the model presented by Müller et al. [116] introduced some novelty yet again. The damping coefficient, similarly to the stiffness one, is in the interval $d \in [0; 1]$. Having the global motion defined by the linear and angular velocities respectively:

$$\begin{aligned} \mathbf{v}_{lin} &= \frac{\sum_i v_i m_i}{\sum_i m_i}, \\ \mathbf{v}_{rot} &= \omega \times r_i, \end{aligned} \tag{2.50}$$

we can compute the deviation of the velocity $\Delta\mathbf{v}_i = \mathbf{v}_{lin} + \mathbf{v}_{rot} - \mathbf{v}_i$. Finally, the damping coefficient can be used to damp these deviations, giving the final velocity $\mathbf{v}_i = \mathbf{v}_i + d\Delta\mathbf{v}_i$.

Deformable Body Improvements

Since we are interested in the *deformable* body behavior we present several constraint functions which control such body properties.

Stretch and Bend Control. A first good example is a constraint function introduced to control the stretching and bending of the material:

$$\mathcal{C}_{stretch}(\mathbf{p}_1, \mathbf{p}_2) = |\mathbf{p}_1 - \mathbf{p}_2| - l_0, \tag{2.51}$$

$$\mathcal{C}_{bend}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4) = acos\left(\frac{(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)}{|(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)|} \cdot \frac{(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_4 - \mathbf{p}_1)}{|(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_4 - \mathbf{p}_1)|}\right) - \varphi_0. \tag{2.52}$$

In these equations l_0 is the initial length of an edge and φ_0 is the initial dihedral angle between two triangles ($\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ and $\mathbf{p}_1$, $\mathbf{p}_4$, $\mathbf{p}_2$) taken into account, as depicted in Fig. 2.5.

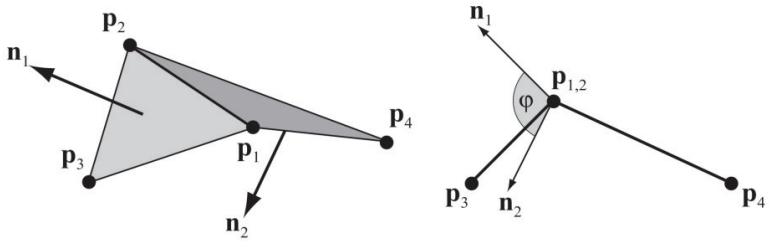


FIGURE 2.5: Two triangles undergoing a bending deformation [116].

In case of cloth simulation one can use a constraint function for cloth balloon (instead of a volumetric mesh) to model a 3D object. Here the t_1^i, t_2^i and t_3^i indices stand for the vertices belonging to a triangle i:

$$\mathcal{C}(\mathbf{p}_1, ..., \mathbf{p}_N) = \left(\sum_{i=1}^{n_{triangles}} (\mathbf{p}_{t_1^i} \times \mathbf{p}_{t_2^i}) \cdot \mathbf{p}_{t_3^i}\right) - k_p V_0, \tag{2.53}$$

where k_p is the pressure constant and V_0 is the initial volume of a balloon.

Volume Control. Another interesting behavior, which can be modeled with PBD, this time using a full 3D object built of tetrahedra, is volume conservation. The constraint function in case of tetrahedral meshes to conserve volume takes the form of the difference of the current and initial volume:

$$\mathcal{C}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4) = \frac{1}{6}(\mathbf{p}_{2,1} \times \mathbf{p}_{3,1}) \cdot \mathbf{p}_{4,1} - V_0. \tag{2.54}$$

An example of Position-Based simulation on a cloth balloon mesh under pressure is shown in Fig. 2.6. The method uses the stretch, bend and volume constraint functions presented above.

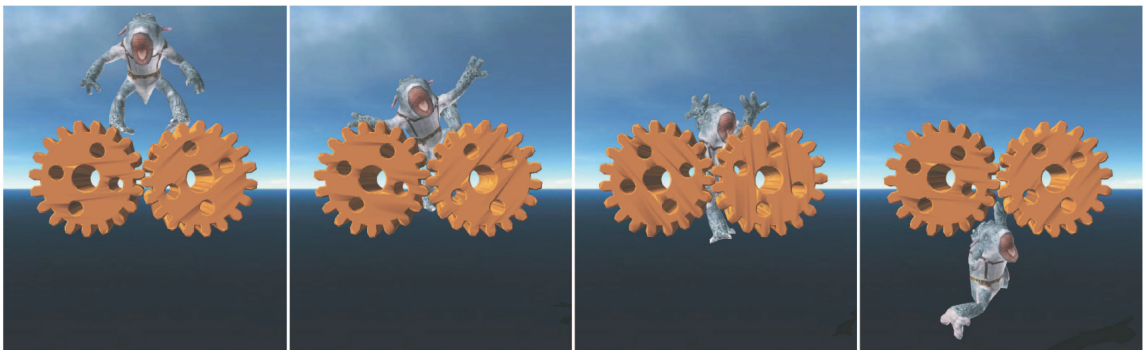


FIGURE 2.6: Cloth character simulated under pressure using PBD [116].

Conclusions

To conclude, Position-Based Dynamics method allows us to manipulate the objects directly during a simulation. It is a stable and controllable method, perfect to use in virtual environments, where user interactions are applied. However, the plausibility of this method can be achieved only visually. The stiffness in PBD, even if incorporates parameters like Young's modulus into the system, is not dependent solely on its formulation - it strongly depends on the number of steps performed by the solver and the size of the time step itself. Another drawback of the method is the fact, that it does not converge to a certain solution when the mesh resolution is increased. For these reasons the PBD can only be used in an approximate simulation of deformable bodies.

2.3.3 Meshless Deformations

Another simulation method used to model behavior of deformable solids, also introduced by Müller et al. [115] and partly used to develop the Position-Based Dynamics method afterward, is called meshless deformation method. As the article's name indicates, the method is based on the shape matching technique, where the energies and forces from the classical approach are replaced by geometric constraints and distances from current to goal positions respectively. The method is suitable to simulate multiple deformable bodies, but more for gaming purposes due to the lack of accuracy of the simulation. Such a method is useful especially in simulation of fluids, but also to model plasticity of the material.

Each object in the meshless simulation is represented by a point-cloud instead of a classical mesh (being points interconnected with some of their neighbors). The extended shape matching technique allows us to model the elasticity of a material by pulling the points towards the goal configuration. In other words the goal positions are determined by use of the shape matching of an undeformed rest state with the current deformed state of the point cloud. In addition, even thought the simulation is handled by the explicit-based integration - the system is unconditionally stable handling even extremely big deformations and allowing the objects to partially recover from such a configuration, like in Fig. 2.7.

Shape Matching Function

The stated problem in the meshless deformation method is to find the rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$ and translation vectors $\mathbf{b}_0$ and $\mathbf{b}$, for two sets of points $\mathbf{x}_i^0$ (initial positions) and

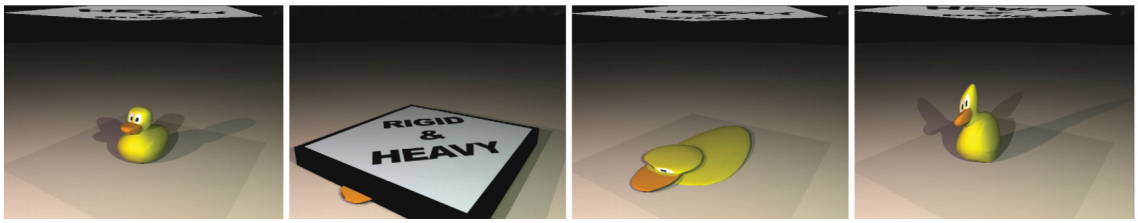


FIGURE 2.7: A duck model squeezed by a heavy object and recovered after the big deformation [115].

$\mathbf{x}_i$ (deformed positions) of masses m_i, to minimize the following equation:

$$\sum_i m_i(\mathbf{R}(\mathbf{x}_i^0 - \mathbf{b}_0) + \mathbf{b} - \mathbf{x}_i)^2. \tag{2.55}$$

The choice for the translation vectors is simple - the center of mass of the deformed ($\mathbf{x}_{cm}$) and undeformed ($\mathbf{x}_{cm}^0$) state will be the minimal values and they read:

$$\mathbf{b}_0 = \mathbf{x}_{cm}^0 = \frac{\sum_i m_i \mathbf{x}_i^0}{\sum_i m_i}, \quad \mathbf{b} = \mathbf{x}_{cm} = \frac{\sum_i m_i \mathbf{x}_i}{\sum_i m_i} \tag{2.56}$$

The rotational matrix $\mathbf{R}$ however is slightly more complicated to compute. It is based on a computation of a linear transformation $\mathbf{A} \in \mathbb{R}^{3\times3}$ instead of the optimal rotational matrix first. The resulting rotational matrix R has a form:

$$\mathbf{R} = \mathbf{A}_{pq}\mathbf{S}^{-1} \tag{2.57}$$

It is composed of two matrices, where $\mathbf{S} = \sqrt{\mathbf{A}_{pq}^T \mathbf{A}_{pq}}$ and

$$\mathbf{A} = \left(\sum_i m_i \mathbf{p}_i \mathbf{q}_i^T\right)\left(\sum_i m_i \mathbf{q}_i \mathbf{q}_i^T\right)^{-1} = \mathbf{A}_{pq}\mathbf{A}_{qq}, \tag{2.58}$$

with the terms $\mathbf{p}$ and $\mathbf{q}$ being respectively the relative locations of the deformed and undeformed state $\mathbf{x}$ with respect to its center of mass. For the detailed explanation of the computation we direct the reader to the article itself [115]. Finally, the goal positions $\mathbf{g}_i$ computed with use of the optimal rotational matrix $\mathbf{R}$ and translational vectors $\mathbf{b}_0$ and $\mathbf{b}$ are:

$$\mathbf{g}_i = \mathbf{R}(\mathbf{x}_i^0 - \mathbf{x}_{cm}^0) + \mathbf{x}_{cm}. \tag{2.59}$$

These positions are used in specially designed explicit-based integration scheme, as we show further on.

Linear and Quadratic Terms

Additionally to the basic shape matching function the quadratic and linear terms are introduced to allow us to model different deformations instead of only small deviations from the rigid shape. First of all, for the linear deformations instead of using the matrix $\mathbf{R}$ from Eq. (2.59), the authors propose to use a modified matrix of a form $\beta\mathbf{A} + (1-\beta)\mathbf{R}$, with β as an additional control parameter. Second of all, the matrix $\mathbf{A}$ is divided by $\sqrt[3]{det(\mathbf{A})}$ to ensure that the determinant of the matrix $\mathbf{A}$ is equal to 1 and therefore the volume of the sample is conserved. And third of all, since the linear transformations can only ensure the

correct behavior in shear and stretch deformations, the quadratic deformations are introduced in the model as well to simulate twist and bending. Similarly to the method of the shape matching function presented above, the authors define a quadratic transformation, this time of a form:

$$\mathbf{g}_i = [\mathbf{AQM}]\tilde{\mathbf{q}}_i. \tag{2.60}$$

They introduce the following quadratic vectors and matrices of respective sizes:

$$\begin{aligned}
&\mathbf{g}_i \in \mathbb{R}^3,\\
&\tilde{\mathbf{q}} = [q_x,\ q_y,\ q_z,\ q_x^2,\ q_y^2,\ q_z^2,\ q_xq_y,\ q_yq_z,\ q_zq_x] \in \mathbb{R}^9,\\
&\mathbf{A},\ \mathbf{Q},\ \mathbf{M} \in \mathbb{R}^{3\times 3},\\
&\tilde{\mathbf{A}} = [\mathbf{AQM}] \in \mathbb{R}^{3\times 9}\\
&\tilde{\mathbf{R}} = [\mathbf{R}\ \mathbf{0}\ \mathbf{0}] \in \mathbb{R}^{3\times 9}.
\end{aligned} \tag{2.61}$$

The matrix $\mathbf{A}$ is holding the information for the linear deformation, matrix $\mathbf{Q}$ the quadratic terms and matrix $\mathbf{M}$ - mixed terms. The solution of minimization term $\sum_i m_i(\tilde{\mathbf{A}}\tilde{\mathbf{q}}_i - \mathbf{p}_i)^2$ is the optimal quadratic transformation matrix:

$$\tilde{\mathbf{A}} = \left(\sum_i m_i\mathbf{p}_i\tilde{\mathbf{q}}_i^T\right)\left(\sum_i m_i\tilde{\mathbf{q}}_i\tilde{\mathbf{q}}_i^T\right)^{-1} = \tilde{\mathbf{A}}_{pq}\tilde{\mathbf{A}}_{qq}. \tag{2.62}$$

Adequately to the linear case, the goal shape is computed with the term $\beta\tilde{\mathbf{A}} + (1-\beta)\tilde{\mathbf{R}}$.

Improved Explicit-Based Integration

As we introduced the basic integration methods in Ch. 2.2, we can directly introduce the integration proposed by Müller et al. [115] as a modification of the semi-implicit scheme (Ch. 2.2.3) with an additional term added to the $\mathbf{v}_{n+1}$:

$$\begin{cases}
\mathbf{v}_{n+1} = \mathbf{v}_n + \Delta t\ \mathbf{F}(x_n, t_n) + k\dfrac{\mathbf{g}_n - \mathbf{x}_n}{\Delta t}\\
\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t\ \mathbf{v}_{n+1}
\end{cases} \tag{2.63}$$

This model uses the knowledge of the goal positions $\mathbf{g}_n$ to not overshoot the calculations. Similarly to the PBD method, the stiffness term $k \in [0; 1]$. The function $\mathbf{F}$ here has the following form: $\mathbf{F}(\mathbf{x}_n, t_n) = \mathbf{F}_{ext}/m$, where $\mathbf{F}_{ext}$ are the external forces.

Method Extensions

The meshless deformation method was further improved by Henriques et al. [69]. First of all they show the problem of volume preservation without additional area-preserving method. In fact the volume preservation causes the body to abnormally grow face-wise at a big deformation, for example like pushing a cube into a wall. The authors suggest then several area-preserving methods to fix this problem, including their three improved ideas based on:

1. scaling the face back to a particular size if the area exceeds some predefined threshold;

2. constrain the magnitude of $\mathbf{A}$ matrix when the predefined threshold is reached. There are two sub-solutions proposed:

- more complex approach with use of the rotation matrix **R** and a coefficient γ derived from the quadratic term;
- simpler approach with the γ parameter simply defined as the predefined threshold divided by the Frobenius norm of matrix **A**.

As an additional visual effect the authors suggest a use of 'soft' treatment of these methods. As constraining the deformation at a threshold causes the object to rapidly stop deforming, they introduced gradual treatment of the area preservation methods. Henriques et al. improve their method once again presenting it in their short paper [70] finally stating that the most promising approach is to limit the deformation by increasing the rigidity of the object. Additionally they use a weighting function, which is preventing discontinuities in the point cloud.

Other meshless simulation methods used for deformable solids and thin-shells, simulating fractures and elastic deformations include work of Guo et al. [67], and for the surgical simulation there are also several meshless methods built on the Element Free Galerkin (EFG) method [72, 73, 83].

2.3.4 Summary

Regardless such an extensive variety of the simulation methods we still have a lot to work on: each of the methods is suitable only for particular application and it differs depending on the type of material (cloth, soft tissue, solid, liquid), deformation (plastic, elastic, linear, etc.), purpose (surgery, real-time, interactive simulation) and many different parameters to be used. Table 2.1 presents a small summary of each method usage purpose, its main advantages and limitations.

Finite Element Method is the most accurate of all the presented simulation techniques and gives a perfect standpoint for any comparisons. Using appropriate constitutive laws we can obtain results, which do not leave room for errors in comparison with experiments performed on real tissues. However, the biggest drawback of this approach is the time of execution of a dynamic simulation. It is frequently made up for with massive precomputations to mitigate the workload. Such approaches may use parallelization techniques to improve the time of the execution. Another problem arises when one wants to perform topological modifications during an ongoing simulation - this forces the classical FEM to recompute boundary conditions for the new topology (necessity of re-doing all the precomputations), which again increases the computation time. Also, as we have discussed, any real-time FEM improvements usually simplify the constitutive laws, which has a direct impact on the method's accuracy.

Position Based Dynamics offers real time simulation opportunities of deformable bodies and proves to be useful in computer graphics thanks to its simplicity, stability, controllability and easy application of the constraint functions. However, we need to take into account that this method is relatively new and only recent work start to show usability of PBD in simulation of deformable objects, where accuracy plays any role as one of the requirements. Main issue of PBD in the context of real tissue simulation is thus the fact, that it is based on the probability and not actual physical laws. This fact makes PBD more suitable for general use in virtual environments, but not simulations which require strong physical basis of the continuum mechanics.

The efficient and unconditionally stable meshless deformations provide many opportunities for interactive simulation without the need of precomputations. Again, as in the case of PBD simulation method, the meshless deformations are not physically motivated,

Method	Purpose	Advantages	Drawbacks
FEM	Predicting behavior of deformable bodies	Various constitutive laws to model body behavior; accuracy.	Time of execution; difficulty to perform topological modifications during an ongoing simulation.
XFEM	Interactive modeling of cracks and fractures	Adapted to model cutting operations.	Unable to model very curved surfaces.
CR FEM	Medical simulators	Recovery of objective nonlinear models	Available only for strains $< 1\%$.
MT	Real tissue modeling	Better performance than classical FEM (precomputed stress tensors).	No rotational invariance in classical MT; computational complexity.
PBD	Rigid and deformable body modeling	Fast execution; easy implementation; direct control over the simulated body; stability.	No physical basis resulting in lack of accuracy; no convergence of the solution upon mesh refinement; time step and iteration - dependent stiffness.
Meshless	Elastic deformable solids	Multiple body simulation; handling extremely large deformations.	No physical basis.

TABLE 2.1: Summary of the simulation techniques: gathering of the main purpose of the method and its advantages and disadvantages.

so the use of this technique is limited mainly to the gaming environment.

There are more and more methods being more flexible and adjustable, but a fully generic method, which allows real-time simulation and offers accuracy of modeling complex materials, still doesn't exist. For the reasons mentioned above we decided to turn our attention to yet another simulation technique with interesting development and application history: Mass-Spring System (MSS). In the process of studying the MSS simulation technique we chose to use FEM as an accurate method to compare our proposed contributions with alongside with any available analytical, physically-based results. In the next chapter we present in detail the MSS method and the related research.

2.4 Conclusions

In this chapter we introduced the basic notions of the mechanics including the continuum mechanics and well known integration schemes, which built a basis for understanding the existing simulation techniques. The pros and cons of these integration schemes make it easier to meet requirements of a simulation, where the stability and size of a time step play the biggest role.

After that we have presented the most known simulation methods used to model deformable bodies in general and we have discussed their application in the computer graphics field. Among them we want to highlight the FEM method as one used most commonly for medical simulation where the accuracy and application of variety of physical laws in form of constitutive functions are the most important.

We have also shown that in the topic of deformable body simulation there are various ways to deal with the problems posed by the physical parameters of soft tissues. The existing techniques are adapted for specific use by every research work to create a wide range on simulation methods to choose from, each being able to model different types of behavior, material and phenomena.

In the next chapter we want to look closer at Mass-Spring System - our simulation method chosen to deal with the real time simulation of soft tissues. We present in detail what MSS challenges us with and how it can be used to serve our purpose of application to medical simulation best.

Chapter 3

Mass Spring System (MSS)

Mass Spring System (MSS) is a fairly simple and intuitive model, incorporating our knowledge about physical laws based on the continuum mechanics into uncomplicated system of point-masses interconnected by springs. It discretizes complex shapes into sets of particles connected together by a set of springs. The behavior of MSS is then evaluated as interactions between the particles modeled by the springs. Then a simulation loop (presented in Ch. 2.2) is used to find new positions of the particles due to the springs' and external forces. The relation of the acceleration of point-masses to the reacting spring forces is used for this purpose. The local deformations of springs result then in the global deformation of the object.

Despite some limitations of the model coming usually from its simplicity (which we discuss in this section further on), MSS are widely used in computer graphics for simulation of clothes [25, 76] or deformable solids [151, 156]. Indeed, the MSS is easy to implement, provides simple calculations and easily incorporates any topological modifications. MSS is also eagerly used for simulation of soft tissues and has been adapted lately for virtual surgery simulation as a real-time performing model.

Due to all the interests, in this chapter we present the detailed structure of classical MSS. We give the formulation of the interaction forces, which comes from the deformation of springs, and is evaluated by the classical simulation loop based on the Newtonian dynamics. Then we discuss the natural MSS limitations and instabilities. We continue by presenting different types of lattice and topologies used for simulation. We discuss how to correctly construct a MSS using particles and springs of particular properties. Further on we talk about the methods for acquisition of MSS parameters like mass, stiffness and damping. We follow by the topic of constraining MSS behavior *e.g.* ensuring correct volume behavior or volume conservation. In the next sections we talk about MSS non-linearity and topological modifications of MSS lattice. At the end, we present our framework for MSS simulation suitable for topological modifications during simulation: its structure, some improvements and implementation details.

3.1 Classical MSS

To use a classical MSS to simulate the elastic behavior of a soft body, we first have to discretize it in a set of particles p_i, with $i \in \mathbb{N}$, of mass m_i, which are connected to one another by springs of a stiffness constant k and initial length l_0. Then the deformation of the springs gives us the deformation of the object. The length of a spring at an elongation δ is then noted $l = l_0 + \delta$. Knowing this, we can define the energy of a spring deformation as:

$$W_{spring}(\delta) = \frac{1}{2}k(l - l_0)^2 = \frac{1}{2}k\delta^2 \tag{3.1}$$

and the force due to this deformation as:

$$\mathbf{F}_{spring}(\delta) = -\frac{\partial W_{spring}(\delta)}{\partial \delta} = -k(l_0 - l) = -k\delta \tag{3.2}$$

In case of a single spring connecting particles p_i and p_j, when a damping d is incorporated into the MSS, the force of a deformed spring with stiffness k_{ij} is then redefined to:

$$\mathbf{F}_{ij}(\delta) = -k_{ij}\delta - d(\mathbf{v}_j - \mathbf{v}_i), \tag{3.3}$$

with $\mathbf{v}_i$ and $\mathbf{v}_j$ being relatively the velocities of particles p_i, p_j.

3.2 MSS Instabilities and Natural Limitations

In this part we present the limitations and natural instabilities of MSS. We present the instances in which they occur and discuss the existing methods used to deal with them. To do that we focus on a MSS formed by a lattice of cubical elements. Each cubical element is composed of 8 particles placed at the vertices. Then Baudet and Zhao [12, 170] have suggested to link these particles together with springs placed on the 12 edges of an element and by springs placed on the 4 inner diagonals of an element.

3.2.1 Natural Limitation of the Poisson Ratio

Let us consider a material with V_0 and V being respectively its initial and current volume.

Linear Assumption

Due to the use of linear springs, the mechanical behavior of the material is linear and follows Hooke's law with a linear relationship [96] between the stress $\boldsymbol{\sigma}$ applied on the material and resulting strain $\boldsymbol{\epsilon}$ with the relation defined in Sec. 2.3 Eq. (2.6). For small deformations, we can use the linearized strain tensor of Green-Lagrange (Eq. (2.8)). Furthermore, the stress tensor $\boldsymbol{\sigma}$ represents the quantity conjugate to the deformation $\boldsymbol{\epsilon}$ in the sense that, when $\boldsymbol{\epsilon}$ varies by $d\boldsymbol{\epsilon}$, the change of energy per unit volume is $d\widetilde{W} = \boldsymbol{\sigma}_{ij}\ d\boldsymbol{\epsilon}_{ij}$. Therefore the expression of the tensor stress is also given by:

$$\boldsymbol{\sigma}_{ij} = \frac{\partial \widetilde{W}}{\partial \boldsymbol{\epsilon}_{ij}}. \tag{3.4}$$

The elastic energy density $\widetilde{W}$ is referred to the unit volume of either deformed or undeformed material, the difference being of higher order in $\boldsymbol{\epsilon}$. The very existence of the energy implies the integrability relations:

$$C_{ijkl} = \frac{\partial \boldsymbol{\sigma}_{ij}}{\partial \boldsymbol{\epsilon}_{kl}} = \frac{\partial \boldsymbol{\sigma}_{kl}}{\partial \boldsymbol{\epsilon}_{ij}} = \frac{\partial^2 \widetilde{W}}{\partial \boldsymbol{\epsilon}_{ij}\, \partial \boldsymbol{\epsilon}_{kl}}. \tag{3.5}$$

The energy per unit volume is defined by the following quadratic relation, with W the energy:

$$\widetilde{W} = \frac{W}{V_0} = \frac{1}{2}\, C_{ijkl}\, \boldsymbol{\epsilon}_{ij}\, \boldsymbol{\epsilon}_{kl}. \tag{3.6}$$

Then, knowing from Cauchy's second law of motion [153] that both $\boldsymbol{\sigma}$ and $\boldsymbol{\epsilon}$ are symmetric, second-rank tensors (*i.e.* $\boldsymbol{\sigma}_{ij} = \boldsymbol{\sigma}_{ji}$ and $\boldsymbol{\epsilon}_{ij} = \boldsymbol{\epsilon}_{ji}$), the elasticity tensor $\mathbf{C}$ initially having $3^4 = 81$ coefficients in 3D is firstly reduced to $6^2 = 36$ coefficients (with $C_{ijkl} = C_{jikl} = C_{ijlk}$) and then reduced to $6 + 6 \times 5/2 = 21$ coefficients by taking into account the integrability relation $C_{ijkl} = C_{kjli}$ (which we explain further on in Eq. (3.12)).

Cubic Symmetry

Considering a material decomposed in cubical elements, the geometrical discretization introduces some symmetries in the system simplifying the formulation of the energy. Indeed, the materials with cubic symmetry are invariant both under $\pi/2$ rotation in a properly chosen frame (x, y, z) and under reflections.

- From the rotational symmetry, we can deduce that coefficients C_{ijkl} are the same for the indices repeating in the same order, like $C_{xyxy} = C_{xzxz} = C_{yzyz}$.
- The reflection property ($x \to -x$, $y \to -y$, and $z \to -z$) gives us interchangeable relations, where $\boldsymbol{\epsilon}_{xy}$ or $\boldsymbol{\epsilon}_{yz}$ are interchangeable respectively with $-\boldsymbol{\epsilon}_{xy}$ and $-\boldsymbol{\epsilon}_{yz}$; involving the fact that we have $C_{ijkl} = 0$ for the components $\boldsymbol{\epsilon}_{ij}\ \boldsymbol{\epsilon}_{kl}$ of the elastic energy, where one of the indices $\{x, y, z\}$ repeats an odd number of times.

Therefore, under cubic symmetry, the energy density of Eq. (3.6) can be rewritten to:

$$\begin{aligned}\widetilde{W}_{cube} = \frac{W_{cube}}{V_0} = \frac{1}{2} C_{xxxx}(\boldsymbol{\epsilon}_{xx}^2 + \boldsymbol{\epsilon}_{yy}^2 + \boldsymbol{\epsilon}_{zz}^2) \\ + C_{xxyy}(\boldsymbol{\epsilon}_{xx}\boldsymbol{\epsilon}_{yy} + \boldsymbol{\epsilon}_{xx}\boldsymbol{\epsilon}_{zz} + \boldsymbol{\epsilon}_{yy}\boldsymbol{\epsilon}_{zz}) + 2\, C_{xyxy}(\boldsymbol{\epsilon}_{xy}^2 + \boldsymbol{\epsilon}_{xz}^2 + \boldsymbol{\epsilon}_{yz}^2).\end{aligned} \tag{3.7}$$

Isotropic Assumption

Considering an isotropic elastic material, the elastic modulus $\mathbf{C}$ is also reduced to only 2 independent parameters. We can choose between Lamé constants (denoted λ, μ) and the pair Young's modulus and Poisson's ratio (denoted E, ν). Moreover, the two latter parameters can be expressed in terms of the two former ones by using:

$$\begin{aligned} E &= \frac{\mu\,(3\lambda + 2\mu)}{\lambda + \mu}, \\ \nu &= \frac{\lambda}{2(\lambda + \mu)}. \end{aligned} \tag{3.8}$$

Therefore, considering the Lamé coefficients, the density energy of equation (3.6) can be reformulated into

$$\begin{aligned} \widetilde{W}_{iso} &= \frac{W_{iso}}{V_0} = \frac{\lambda}{2}(Tr(\boldsymbol{\epsilon}))^2 + \mu\, Tr(\boldsymbol{\epsilon}^2) \\ &= (\frac{\lambda}{2} + \mu)(\boldsymbol{\epsilon}_{xx}^2 + \boldsymbol{\epsilon}_{yy}^2 + \boldsymbol{\epsilon}_{zz}^2) \\ &+ \lambda\,(\boldsymbol{\epsilon}_{xx}\,\boldsymbol{\epsilon}_{yy} + \boldsymbol{\epsilon}_{xx}\,\boldsymbol{\epsilon}_{zz} + \boldsymbol{\epsilon}_{yy}\,\boldsymbol{\epsilon}_{zz}) + 2\,\mu\,(\boldsymbol{\epsilon}_{xy}^2 + \boldsymbol{\epsilon}_{xz}^2 + \boldsymbol{\epsilon}_{yz}^2). \end{aligned} \tag{3.9}$$

We obtain the corresponding definition of the elastic tensor

$$C_{ijkl_{iso}} = \frac{\partial^2 \widetilde{W}_{iso}}{\partial \boldsymbol{\epsilon}_{ij}\,\partial \boldsymbol{\epsilon}_{kl}} = \lambda\,\delta_{ij}\delta_{kl} + \mu\,(\delta_{ik}\,\delta_{jl} + \delta_{il}\,\delta_{jk}). \tag{3.10}$$

Isotropic cubical MSS. Comparing equations (3.7) and (3.9), materials with cubic symmetry can be considered as isotropic if the elastic coefficients satisfy the following relations:

$$C_{xxxx} = C_{xxyy} + 2\,C_{xyxy},\; C_{xxyy} = \lambda,\; C_{xyxy} = \mu. \tag{3.11}$$

The above is valid for any material with cubic or isotropic symmetry. Finally, the macroscopic elasticity tensor for mass-spring systems being completely symmetric with respect to interchange of its four indices (Saint-Venant, 1855), so where vertices are centers of symmetry, the elastic coefficients must satisfy the further Cauchy relations, involving the Cauchy equality $\lambda = \mu$ from Eq. (3.11):

$$C_{ijkl} = C_{kjli}. \tag{3.12}$$

As a consequence, from Eq. (3.8) we can only define the Young modulus as $E = 5/2\mu$ and the Poisson ratio consistently as $\nu = 1/4$ in 3D deformation of isotropic mass-spring systems. In other words, isotropic cubical - symmetric MSS are unable to simulate materials with a Poisson ratio different than $1/4$.

3.2.2 Natural Wrinkle and Buckling Instability of the MSS

Let us again consider a material modeled by one cubical MSS, where springs are placed on the edges and on inner diagonals of the element. In this configuration, some instabilities appear as wrinkles on the borders of the object for certain compression rates. Moreover, this buckling instability alters the overall behavior by making the system less rigid then it should be according to any analytical solution.

The origin of these instabilities can be easily explained if we consider a simple model composed of 5 particles (noted A, B, C, D, E) connected by 4 springs defined by a stiffness constant k and a rest length l (as illustrated in Fig. 3.1). Particles D and E are fixed. Particle C is free to move in the horizontal plane with a distance denoted x from its initial position. Particles A and B are free to move in the vertical direction from their initial positions (drawn in green). After an identical deformation, their lengths are denoted ζl. Consequently, the lengths of springs AC and BC are defined by $l'(x,\zeta) = \sqrt{(\zeta l)^2 + x^2}$ and the rest state of this system, *i.e.* the state where all the springs have their natural length l, corresponds to $x = 0$ and $\zeta = 1$.

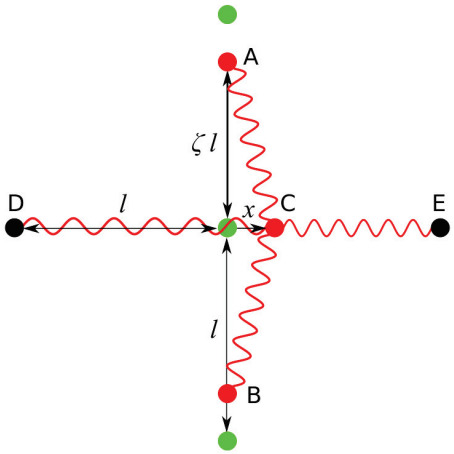


FIGURE 3.1: A MSS composed of 5 particles (noted A, B, C, D, E) connected by 4 springs with stiffness constant k and rest length l. Green points represent the rest positions of particles A, B and C.

The elastic energy $W(x,\zeta)$ of this four-spring MSS is written:

$$W(x,\zeta) = k\left(\left(l'(x,\zeta) - l\right)^2 + x^2\right), \tag{3.13}$$

with a stiffness constant k and an elongation x defined by $(k\ x^2)/2$, from which we can compute the horizontal force $\mathbf{F}_C(x,\zeta)$:

$$\mathbf{F}_C(x,\zeta) = -\frac{\partial W(x,\zeta)}{\partial x} = 2\ k\ x\left(\frac{l}{l'(x,\zeta)} - 2\right). \tag{3.14}$$

Note, that the horizontal force affects the particles A, B and C, but only C can actually move in the horizontal plane.

The equilibrium equation $\mathbf{F}_C(x,\zeta) = 0$ has 3 roots: $x = 0$ and $x = \pm\ l\ \sqrt{\frac{1}{4} - \zeta^2}$ corresponding to $l/l'(x,\zeta) - 2 = 0$. Note that changing the sign of ζ results simply in interchanging the positions of points A and B, leaving $l'(x,\zeta)$, $W(x,\zeta)$, $\mathbf{F}_C(x,\zeta)$ unchanged.

- In the range $|\zeta| > 1/2$, when $l/l'(x,\zeta) - 2 < 0$, the last two roots are imaginary, whereas the only physical root $x = 0$ corresponds to a stable position. In this configuration, the sign of $\mathbf{F}_C(x,\zeta)$ being opposite that of x.
- In the range $|\zeta| < 1/2$, when $l/l'(x,\zeta) - 2 > 0$, the last two roots are real and stable. In the same time $x = 0$ is an unstable root and $\mathbf{F}_C(x,\zeta)$ having the same sign as x.

Taking into account all these cases, the stable positions of point C are thus:

$$x = \begin{cases} 0 & \text{if } |\zeta| > 1/2 \\ \pm l\sqrt{\frac{1}{4} - \zeta^2} & \text{if } |\zeta| < 1/2 \end{cases} \tag{3.15}$$

Therefore the system undergoes a bifurcation at $\zeta = \pm 1/2$. The aligned configuration of A, B and C particles (so with $x = 0$) becomes unstable when $|\zeta| < 1/2$ and changes into a configuration with $x \neq 0$, *which is the source of wrinkles*. Remarkably, by inserting solution 3.15 into the definition of $l'(x,\zeta)$, we find $l'(x,\zeta) = l/2$ within the whole buckling range $|\zeta| < 1/2$, whereas $l'(x,\zeta) = l|\zeta|$ when $|\zeta| > 1/2$.

Now, if we do not observe the particle C, how shall we know that there is something going on? Let us assume a stable position of the particle C defined by Eq. (3.15) and consider the resulting elastic energy as a function of ζ:

$$W_{\rm eq}(\zeta) = \begin{cases} W(x=0,\zeta) & \text{if } |\zeta|>1/2, \\ W(x=\pm l\sqrt{\frac{1}{4}-\zeta^2},\zeta) & \text{if } |\zeta|<1/2 \end{cases} \tag{3.16}$$

$$= \begin{cases} k\,l^2(|\zeta|-1)^2 & \text{if } |\zeta|>1/2, \\ k\,l^2\left(\frac{1}{2}-\zeta^2\right) & \text{if } |\zeta|<1/2 \end{cases} \tag{3.17}$$

The vertical component of the force that the springs exert on particles A and B are given by:

$$\mathbf{F}_A(\zeta) = -\mathbf{F}_B(\zeta) = -\frac{1}{2\,l}\,\frac{\partial W_{\rm eq}(\zeta)}{\partial \zeta} \tag{3.18}$$

$$= \begin{cases} k\,l(\frac{|\zeta|}{\zeta}-\zeta) & \text{if } |\zeta|>1/2 \\ k\,l\,\zeta & \text{if } |\zeta|<1/2 \end{cases}$$

In Fig. 3.2, the functional form of $\mathbf{F}_A(\zeta)$ exhibits kinks at $\zeta=\pm 1/2$. Buckling can result in hysteretic behavior. Consider the effect of a compressive force of an increasing magnitude, as displayed with dashed arrows. The system will first respond with a compression $\zeta = 1-\mathbf{F}_A(\zeta)/k\,l$ until the critical buckling force $\mathbf{F}_A(\zeta)=k\,l/2$ is reached at $\zeta=1/2$, then suddenly the particles will distort and jump to an inverted configuration with $\zeta=-3/2$, with particle A below particle B, then further deform according to $\zeta=-1-\mathbf{F}_A(\zeta)/k\,l$. On the way back buckling will initiate at $\zeta=-1/2$, with $\mathbf{F}_A(\zeta)=-k\,l/2$ and the system will jump to $\zeta=1/2$.

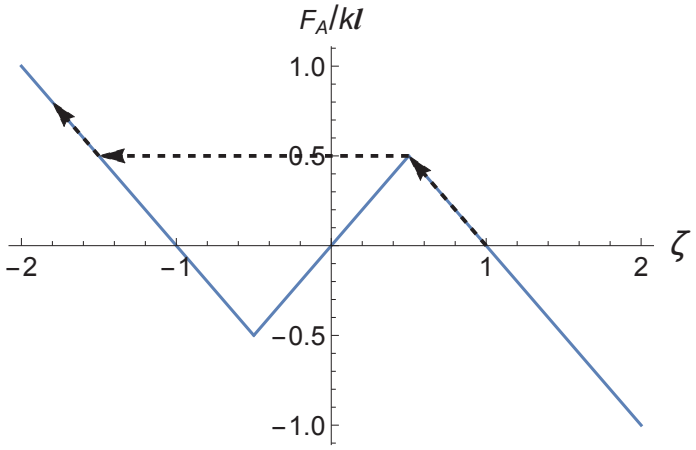


FIGURE 3.2: Curve: normalized vertical force as a function of extension ζ according to Eq. (3.18). Dashed arrows: trajectory under progressive compression. Buckling occurs for $-1/2<\zeta<1/2$.

Note that the precise buckling threshold value and the slope of the force in the buckling domain are peculiarities of the MSS system. They can be changed for instance by making the horizontal springs differ from the vertical ones or by adding a spring between A and B particles. However, the general conclusion persists: the buckled system shows

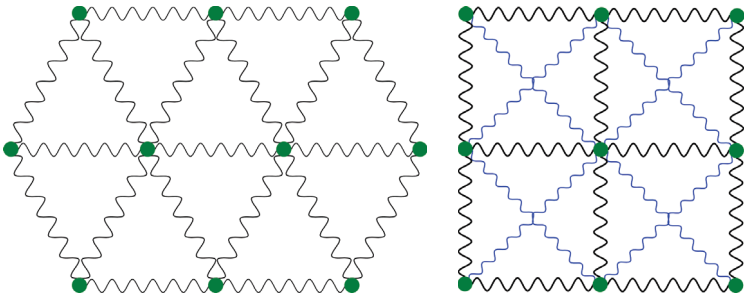
less rigidity than it did before buckling. Consequently, in addition to superimposing extra local wiggling, this instability alters the overall deformation of elastic spring system.

3.3 Types of Lattice

To apply the physical laws defined above we need to structure our model and know how to connect the necessary information about the physical properties (like mass, stiffness, spring length, *etc.*) to the actual mesh components. For this purpose let us study first the existing types of mesh used for simulation with, but not only, the MSS model.

3.3.1 Classical Meshes

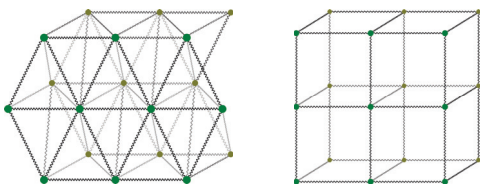
The most commonly used types of lattice in 2D MSS are triangular (Fig. 3.3 (a)) and, in some rare cases, quadrangular ones (Fig. 3.3 (b)). The triangular meshes are usually used thanks to the simplicity in constructing them and no need for any additional springs, since a mesh constructed of triangles is already stable. These meshes are generally used in cloth simulation. Moreover, they provide simple solutions to problems like *e.g.* face area preservation or refinement. In 2D quadrangular meshes there are usually two types of springs: edge springs (depicted as the black springs in Fig. 3.3) and face diagonal springs (drawn in blue). The quadrangular mesh requires adding the face-diagonal springs to help preserving its shape and stabilize it. The green nodes - vertices, which hold information about their masses and positions usually hold also information about their velocities and the springs they are connected to (for example in a form of a list). The springs are defined by information like current and initial length, stiffness and damping.



(a) 2D triangular; (b) 2D quadrangular.

FIGURE 3.3: 2D MSS: (a) triangular and (b) quadrangular with blue face diagonal springs.

Adequately, in 3D the most known are tetrahedral (Fig. 3.4 (a)) and hexahedral (Fig. 3.4 (b)) meshes. 3D meshes are used for more complex deformable solids, where the tetrahedral ones are again the preferred type.



(a) 3D tetrahedral; (b) 3D hexahedral.

FIGURE 3.4: Two types of 3D MSS meshes.

Such a mesh does not need to be built of the perfect triangular or quadrangular elements. In fact it is also possible to use a mesh of unstructured geometry (see Fig. (3.5)), which sometimes can give more realistic results of the simulation (some examples of differences between experiments performed on structured and unstructured meshes are presented by Kot et al. [92]). In this case the structured mesh from Fig. 3.5 (left) is more suitable for an anisotropic body simulation, while the mesh from Fig. 3.5 (right) can be used to represent an arbitrary cloth. Additionally such a mesh can be locally refined, which affects the level of detail and accuracy of a simulation.

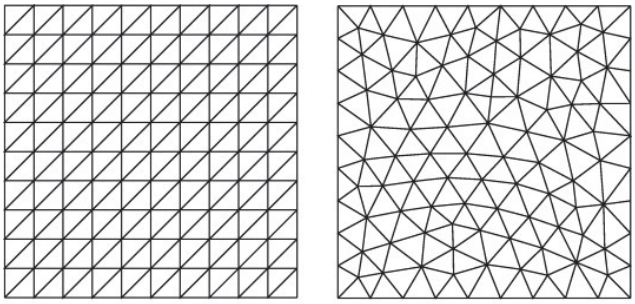


FIGURE 3.5: Structured (left) and unstructured (right) 2D triangular mesh [71].

3.3.2 3D Surface Mesh

Another possibility, which is a variation of 2D mesh in 3D body simulation, is called cloth balloon mesh (see Fig. 3.6) or 3D surface mesh. It is used to describe objects in form of (usually) triangular 3D surface meshes [4, 50]. Such a mesh can be used to simulate 3D objects as literal cloth balloons, also with underlying rigid or deformable structures like bones or muscles. It is important to structure such a mesh carefully: in Fig. 3.6 (left) we can see that the sphere mesh has a closed structure, so it can be easily used as an inflated balloon. On the other hand the mesh of same type can be open Fig. 3.6 (right), as the skull surface, therefore it is not suitable for such a use.

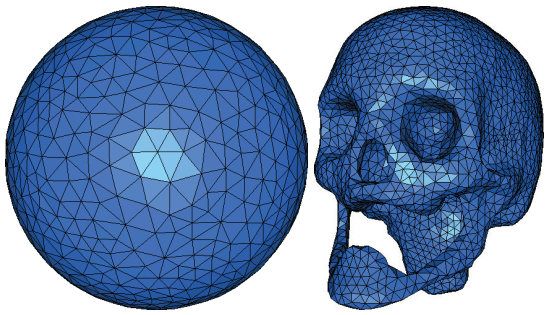


FIGURE 3.6: Sphere-surface (left) and skull-surface (right) meshes [137].

A sub-type of a surface mesh is structured with rigid elements as the inner basis. Due to the special character of this type of lattice one needs to understand its structure. Fig. 3.7 presents a triangular lattice consisting of 6 nodes. The central node p_i is connected to the center point c_i, which is found thanks to the normal to the surface N_i of the initial length l_i. Each of the neighbouring nodes is also connected to this central point. There are two possible ways to construct a surface mesh. One is a radial approach, which is connecting all of the surface nodes to a central point. The second one however, as presented by Arnab

et al. [4], is to re-mesh the lattice for each vertex i to obtain the configuration presented in Fig. 3.7. The tetrahedra created this way allow us to control the underlying volumes, which we describe in more detail in Ch. 3.5.

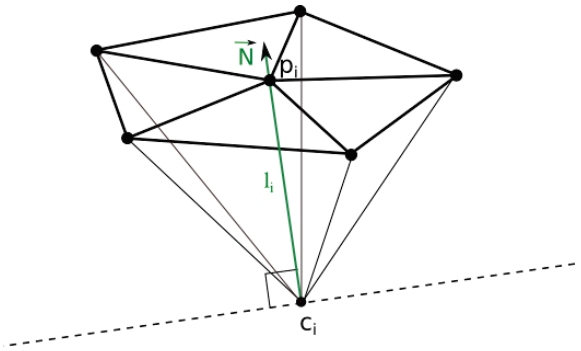


FIGURE 3.7: A 3D surface mesh with normal to the surface N_i and spring of length l_i connecting the node p_i with a central point c_i.

3.3.3 Mixed-Element Mesh

Mixed-element mesh is a type of mesh built of several types of elements. Such a mix can consist of triangular and quadrangular elements in 2D or, for example tetrahedra connected with prisms and pyramids in 3D. Using mixed-element meshes is especially important in the Finite Element Simulation, which is the reason why most of the contributions from literature are developed with FEM in mind. At this point we have to draw a line between conforming and non-conforming meshes. If for every two neighboring elements their faces match exactly we are dealing with a conformal mesh. Otherwise the mesh is called non-conformal. This means that to build a conformal mesh in 2D a single edge cannot have more than two adjacent faces. Similarly, in 3D a single face cannot have more than two adjacent volumes. A simple example of 2D conformal and non-conformal meshes is depicted in Fig. 3.8.

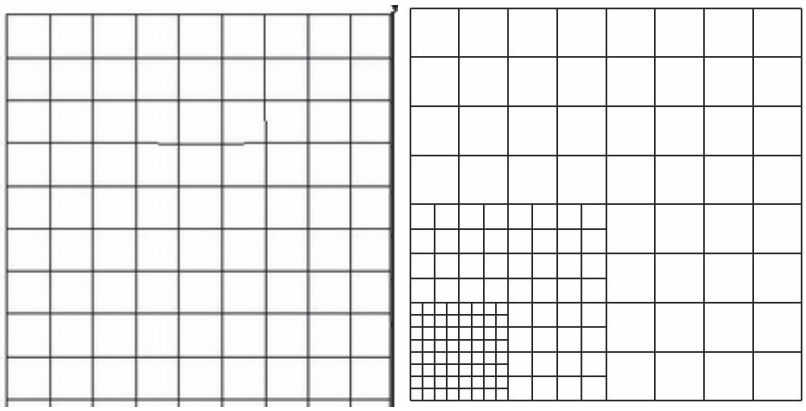


FIGURE 3.8: (left) A conformal mesh; (right) A non-conformal mesh.

Already since 1993 the researchers were trying to find their way to deal in a simple way with mixed-element meshes, like tetrahedra-hexahedra non-conformal mesh idea of [48, 49]. Then, some ideas on creation of conformal meshes started to make their way, like injecting pyramids as the joints between tetrahedra and hexahedra introduced by [132] and [133].

The mixed-element mesh gives significantly better results than just using all-tetrahedral meshes [165]. Additionally, using this type of mesh allows much better performance, since the number of nodes is decreased, while the quality of the mesh remains practically intact if no geometrical changes are applied to the mesh. As an example we can cite the work of [107], where the authors develop a method to generate penta- and hexahedral elements using classification with a graph representation or [164] who convert an all-tetrahedral mesh into a prism-tetrahedral one by gradually inserting prism layers.

3.4 Identification of the Physical Parameters

The particle-based systems are in general very intuitive, but lack accuracy. In addition, the application of physical and constitutive laws into the method is not an easy task. The choice of physical parameters of the springs, that is stiffness and damping coefficients, play an important role in the stability of the system and in its precision.

In the simulation of real deformable bodies, there is no obvious way to identify the parameters responsible for elasticity, plasticity, viscosity and many other attributes, which in theory could be drawn from constitutive laws. Every body exhibits slightly different material properties, which then have to be 'translated' to MSS properties. Over the years the authors tried to assign particular values to masses, MSS stiffness or even damping coefficients. Many authors put up with the topic [84, 99, 119] and presented some of the methods of the parameter acquisition including their novel approaches.

In general we can distinguish two main methods for the identification of the physical parameters to obtain a physically valid MSS: 1) physical experiments and real-model matching; 2) derivation of an analytical solution applicable in the spring coefficients. As long as the second method does not have different examples, the first one includes learning algorithms, visual comparison with the real models (like deformation of clothes) and actual comparison of the *response* of a tissue and simulation model during different deformations. The acquisition of the parameters however is not so straight forward, so we present several of them to have an overview on the research done in this direction.

3.4.1 Masses

First we have the global mass of an object. [47] presented a universal model to compute the vector of masses of MSS. The model, which is used by the vast majority of researchers in simulation of homogeneous materials, is distributed over the different particles of the MSS. That is, considering a 3D object of mass density ρ, the mass m_i of each particle i is defined by

$$m_i = \sum_j \frac{\rho}{8} V_j^c, \tag{3.19}$$

with $\sum_j$ being the sum over all the j cubical elements to which m_i belongs, with volumes defined by V_j^c.

3.4.2 Damping

As far as it goes for the damping coefficient d, we have no doubts how and in which cases it should be used. We consider a simple MSS consisting of two particles of masses m_1 and m_2, with $m = m_1 + m_2$, connected by a spring of stiffness constant k and rest length l_0. As MSS is a system with external forces acting on it, we can study the damping ratio ζ of

complex oscillations within the springs [22]. In that regard MSS can be undamped ($\zeta = 0$) or damped in one of the three following ways:

- **underdamped** - the system is decaying exponentially with oscillation at time t of a form $\exp(i\sqrt{\frac{k}{m}(1-\zeta^2)}t)$, where $\zeta < 1$ and $i = \sqrt{-1}$; in this case we have: $0 < \zeta < 1$;
- **overdamped** - the system is decaying exponentially with no oscillation, with $\zeta > 1$;
- **critically damped** - the system goes to its equilibrium state as fast as possible without any oscillations. Such a behavior is possible when $\zeta = 1$, with the corresponding critical damping coefficient $d = 2\sqrt{km}$

Usually it is the critical damping used in the MSS to avoid unnecessary oscillations and naturally damp the system as fast as possible. However it can be also adjusted (by scaling it with use of the spring length l_0) to handle the same amplitude of oscillations for different mesh resolutions [134, 167]:

$$d = 2\frac{\sqrt{k(m_1 + m_2)}}{l_0}. \tag{3.20}$$

3.4.3 Stiffness

To obtain the stiffness for a simulation of deformable bodies there are generally two approaches: data driven parameter identification and analytical solution based on physical body properties or a constitutive law. Let us present several methods used for this purpose.

Data Driven Parameter Identification

The data-driven approach is to adjust the spring parameters according to some previously acquired set of physical or theoretical data. These information can be, and very often are, taken from visual aids, like videos of body deformation. Having such sets of the goal data it is easier to adjust the MSS accordingly and find the optimal parameters. This method was eagerly used by Louchet et al., who introduced visual comparison combined with evolutionary genetic strategy [101]. The goal was to model isotropic, homogeneous cloth, which consisted of 6 types of springs, each described by 3 parameters, but simplified to 5 parameters for the fabrics: spring stiffness, elongation rate, and 3 pairs of natural lengths.

A genetic algorithm was used by Joukhadar et al., who tried to determine the physical parameters such as viscosity, elasticity or plasticity to describe the object for a dynamic simulation [84]. The MSS used by the authors is built of tetrahedral 3D mesh.

Bianchi et al. in turn used genetic algorithm to teach their MSS the correct behavior by assigning correct stiffness value to the springs [20, 19]. First article was focused on the 2D model, but the second one extended the method to 3D. The reference model was linear elastic Finite Element Method. The model was supposed to learn the correct behavior to simulate soft tissues as a linear elastic material. What is interesting, the authors shown that the parameter distribution in MSS has to be non-homogeneous to model this type of body.

In the cloth simulation, Bhat et al. presented a method to obtain the parameters of different fabrics (like linen, satin or knit) based on some video samples [18]. A simulated annealing method was used to compare the simulation and video data and to optimize the triangular 2D MSS by studying the folds of the material.

Previously mentioned Deussen et al. used a theoretical elastic model as a basis for the comparison and parameter identification [47]. Again, an optimized simulated annealing is used to obtain elastic parameters of the MSS. Instead of obtaining the parameters directly, the behavior of MSS is being optimized over time to obtain the best results of the simulation.

Da Silva et al. presented a method based on the data-driven strategy comparing the model's acceleration to have similar behavior of the MSS and the reference model [144]. The derivation of the parameters is divided into several elements: solving the optimization to compute the stiffness parameters, compute the masses and the damping.

San-Vincente et al. studied the cubical MSS with 3 types of springs (edge, inner and face diagonal) and based their parametrization approach on a tensile test and a non-linear material model [141]. On top of the derived model they decided to add an objective function to optimize the model's parameters but fitting the curve with respect to some reference data, therefore obtaining particular stiffness values for particular experiment cases. They used Neo-Hookean, Mooney-Rivlin, Yeoh, and Ogden models, usually used to model non-linear behavior of rubber-like materials or soft tissues.

Data driven approach is generally well optimized for a particular type of body in a predefined simulation environment and under specific circumstances. Such a setup largely limits the possibilities of simulation, especially if anything shall change in the modeled environment. Unless we want to be limited to such a narrow use of the simulation tool, we need to reach for more generic method which takes into account the physically-based analytical behavior.

Analytical Solutions

Since the soft bodies have particular properties, like elasticity, viscosity, etc., some authors decided to derive the parameters of MSS from the theoretical laws governing these body properties. Some of them are based on theoretical laws, some use simplified approximations and some draw inspiration directly from the Finite Element approach. Let us present the existing methods in a form of a table (Tab. 3.1) and a corresponding list which describes the presented approaches. The table structure allows us to verify the exact formulas, types of simulated tissue and the physical basis on which these were derived. In the second column we listed the authors of the related articles, then the formulations of stiffness followed by the properties of the simulated material. In the last column we note the type of derivation the presented formulas are based on.

1. One of the first attempts in MSS parametrization was done by Wilhelms and Van Gelder [160]. The elastic membrane model which they used divided the structure of MSS into two components: skin MSS and underlying structures MSS (bones - rigid MSS; muscles - deformable MSS). The stiffness k of a skin spring, which belongs to two neighbouring triangles was defined with their respective surface areas A_1 and A_2. The second type of springs was used to connect the skin and the virtual anchor and its stiffness was denoted k_a. Its rest length was $l_0 = 0$, but its stiffness, which was summed over the i incident triangles, was also multiplied by a proportionality constant coefficient c to allow 'sliding' of the skin mesh on the bones and muscles.

2. A year after that Gelder et al. decided to derive an approximation of the stiffness based on the geometry of a mesh [64]. Their formulation evolves around the elasticity law and incorporates Young's modulus E and Poisson's ratio ν into the model.

N°	Author	Stiffness	Material properties	Physical basis
1.	Wilhelms et al. [160]	$k = \frac{A_1 + A_2}{l_0^2}$ $k_a = c \sum_i \frac{A_i}{3}$	2D triangular, elastic membrane	geometrical approximation
2.	Van Gelder et al. [64]	$2D: \quad k_c = \frac{E_2 \sum_e A_e}{\lvert l_0 \rvert^2}$ $3D: \quad k_c = \frac{E \sum_e V_e}{\lvert l_0 \rvert^2}$	2D triangular or 3D tetrahedral, isotropic, linearly elastic	geometrical approximation
3.	Zerbato et al. [167]	$k = \frac{E_1 + E_2}{2} \frac{\sum_i V_i}{l_0^2}$ $+$*genetic algorithm*	real liver tissue	geometrical approximation
4.	Arnab et al. [4]	$k_e = E \sum_{i \in j} \frac{V_j}{l_i^2}$, $k_g = E \sum_{i \in j} \frac{V_j}{2\, l_i^2 (1-\nu)}$.	3D elastic soft tissues	geometrically based model with elasticity properties
5.	Maciel et al. [102]	$k = \frac{k_0 \cos_0 + \sum_i k_i \cos_i}{l_0 + \sum_i l_i \cos_i}$.	elastic linear soft tissues; 3D hexahedral mesh	angular adjustment of the formula of Van Gelder et al.
6.	Lloyd et al. [99]	$k_{tri} = \sum_e Eh \frac{\sqrt{3}}{4}$, $k^e_{rect} = \sum_e \frac{5}{16} hE$, $k^d_{rect} = \frac{7}{16} hE$, $k_{tetra} = \sum_e \frac{2\sqrt{2}}{25} lE$, $k_{tetra2} = \sum_e \frac{2\sqrt{2}}{21} lE\frac{4}{5}$, $k_V = \frac{\sqrt{2}}{84} l^3 E \frac{2}{5}$.	linear elastic material; two last stiffnesses for a volume preserving elastic tetrahedral mesh	based on a reference model and FEM discretization
7.	Baudet et al. [12]	$k_e = \frac{E(j^2(3\nu+2) - i^2)}{4 l_0 h_0 (1+\nu)}$, $k_f = G = \frac{E}{2(1+\nu)}$, $k_{ed} = \frac{E l_0 (4\nu+1)}{8(1+\nu)}$, $k_d = \frac{3 E l_0}{8(1+\nu)}$.	linear elastic material with rectangular and hexahedral mesh	parametrization based on tensile and shearing deformations
8.	Natsupakpong et al. [119]	$k_e = \frac{\lambda}{3} + \frac{\mu}{2}$, $k_f = \left(\frac{5\lambda}{12} + \frac{3\mu}{4}\right)$	linear elastic material; 2D rectangular mesh	approximation of FEM

TABLE 3.1: The stiffness parameters used in the field of simulation of deformable objects.

The formula presented in Tab. 3.1 is a simplified version of a more expansive form, which states for a spring c:

$$k_c = \left(\frac{E_2}{1+\nu}\right)\frac{A_e}{|l_c|^2} + \left(\frac{E_2\nu}{1-\nu^2}\right)\frac{|l_a|^2+|l_b|^2-|l_c|^2}{8A_e}. \tag{3.21}$$

In this equation $E_2 = Eh$ is a parameter defined with Young's modulus multiplied by the thickness h of a material; A_e is the area of a triangle adjacent to the spring c, with a and b being the two other springs and l_a, l_b, l_c their relative lengths. This formulation is a general one, which can be used for triangles of any type. However in their experiments the authors used $\nu = 0$, which simplifies this equation. They decided to use the simple form, since they did not see any significant impact on the simulation when $\nu = 1/4$ was used. The results are studied only visually, so the accuracy of this formulation was not established.

The work of Wilhelms and Van Gelder was used also by several other authors [167, 4, 112]. Mollemans et al. decided to use MSS for maxillofacial surgery purposes where their model consists of bone and tissue nodes called respectively joint and free points. Unlike the free points, the joint points provided an actual connection between the soft tissue part and the bone structure, so they were not allowed to change their positions.

3. The method of Zerbato et al. does not really fall into the method derived from analytical solution, since they used a genetic algorithm to adjust the MSS parameters, however they did use a slightly modified stiffness formula of Van Gelder et al. [64] to model their 3D tetrahedral MSS as a starting point to the simulation. In the formula presented in Tab. 3.1 E_1 and E_2 are the Young moduli of the two vertices the spring connects and V_i are the volumes of i tetrahedra incident to the spring. As we can see, E is simply replaced by the average of E_1 and E_2 terms. The difference between simulated and theoretical E was relatively big (about 9.49%), however the max. absolute error along different axes was measured to be of a $4 \cdot 10^{-4}$ magnitude.

4. Arnab and Raja used a surface mesh with two stiffnesses: k_e based on the linear elasticity law (Young's modulus) and k_g based on the shear modulus (G, dependent on the Young modulus and Poisson ratio). The formula is based on nodes $i \in \{1, ..., N\}$ (with N being the number of vertices) belonging to a triangular element $j \in \{1, ..., M\}$ (where M is the number of triangles in the surface mesh).

5. An analytical expression for stiffness of cubical meshes was introduced by Maciel et al. [102]. They focused on the elastic linear behavior of a simplified MSS called molecular model. Their MSS mesh is built of round masses (molecules) interconnected by edge springs and face-diagonal springs. The computation of the spring stiffness was based on the uniaxial tensile test and it depends on the angular configuration of the springs. The stiffness was defined for $i \in [1;\ N]$, where N is the number of connections sharing an extremity with spring k_0, and $\cos_i$, which is the cosine function of the angle between spring 0 and spring i. It was based of Van Gelder's formula [64], but with the angular adjustment.

6. Lloyd et al. discussed about the previously developed approaches to identify the MSS parameters and introduced their own, more accurate and reference based method for symmetric triangular, rectangular and tetrahedral meshes [99]. They focused on

the stiffness acquisition based on the reference FEM model, that they compare their method to. The stiffness is dependent only on the Young modulus E and the thickness of the plane stress elastic model h (in case of 2D meshes) or spring's length l (in case of the 3D mesh). The formulas for triangles (k_{tri}), rectangles (k^e_{rect} for edge spring and k^d_{rect} for diagonal spring) and tetrahedra (k_{tetra}) are presented in point 6 of Tab. 3.1.

Their formula for the tetrahedral 3D mesh is still used by many researchers [140, 52]. The authors also derived stiffnesses for volume preserving regular tetrahedral meshes. Since the reference FEM model is different for the volume preserving case, they had to derive a new set of equations. The last two formulas presented in the table are: k_{tetra} - the stiffness for the regular force term and k_V - the stiffness parameter for the volume preserving force. The computation is also based on the reference FEM model and more specifically stiffness matrix derived from a volume preserving potential energy.

In Fig. 3.9 we can see a 2D MSS mesh simulation (red) on top of a reference FEM (black). The experiment shows a linear deformation of MSS with use of the stiffness formula k_{tri}. In a case of such a small deformation the difference between MSS and FEM is visible (especially in a shearing experiment) but does not exceed 2%, which is a very good result. Unfortunately, the same method of derivation could not be used for 3D tetrahedral elements and the stiffness was only approximated in this case (k_{tetra} from Tab. 3.1). The results of these experiments (see Fig. 3.10) are somewhat worse, with the largest error being 0.111 cm on a sample of 1 cm radius and 2 cm height in the equilibrium state, which already makes it impossible to use for the precise medical purposes. Additionally there are two assumptions which make the system work well: Poisson's ratio set to $\nu = 1/3$ and an equilateral triangular mesh.

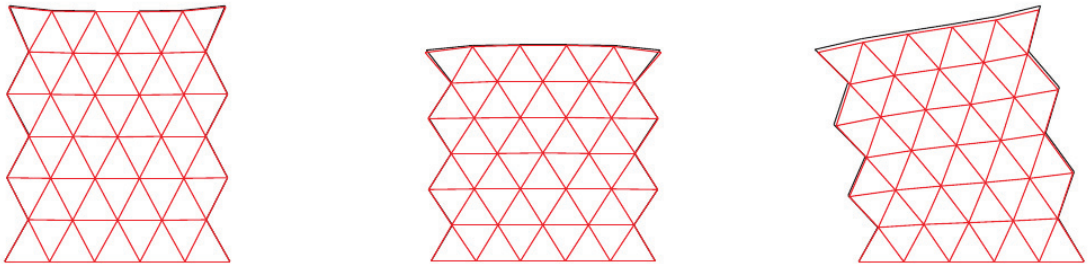


FIGURE 3.9: Linear elastic material simulated with an equilateral triangular mesh. From left to right: stretching, compressing and shearing experiments. MSS is depicted in red and corresponding FEM in black [99].

7. Baudet et al. decided to use 2D quadrangular and 3D cubical MSS model (the latter one with only two types of springs: edge and inner diagonal ones) [12]. The stiffness parameters are then derived in 2D from a shear and tensile tests and then extended to 3D case. To fix the inaccuracies coming from this derivation the authors introduced also corrections to the forces. Baudet et al. formulation of the stiffness evolves around the young modulus E, the Poisson ratio ν and rest length of a spring l_0. For a 2D rectangular element l_0 is the initial width and h_0 the initial height of a rectangle and $(i, j) \in \{l_0, h_0\}^2$, such that $i \neq j$. Similarly, for a 3D symmetric (cubical) element of an edge length l_0, the stiffness formulas are defined by stiffness for edge k_{ed} and inner diagonal k_d springs. The authors noted, that their derivation is valid only if Poisson ratio is equal to $\nu_{2D} = 1/3$ or $\nu_{3D} = 1/4$, so they introduced

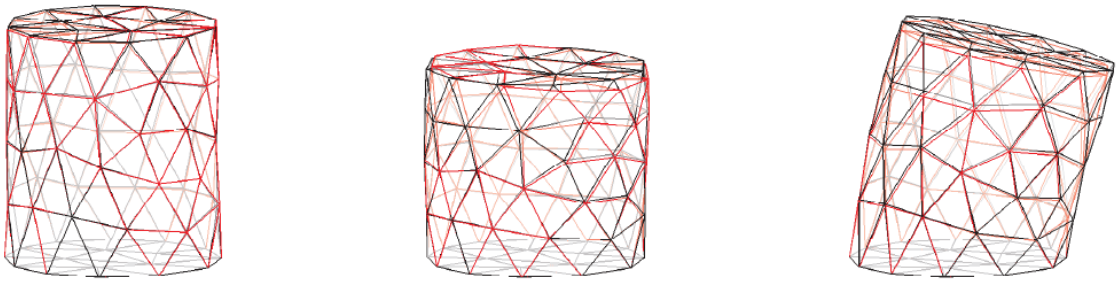


FIGURE 3.10: Linear elastic material simulated with a tetrahedral mesh. From left to right: stretching, compressing and shearing experiments. MSS is depicted in red and corresponding FEM in black [99].

correction forces if different values of ν are used. We talk about this method in detail in Ch. 3.8.2.

8. Natsupakpong et al. derived the linear elastic parameters of MSS (or as they called it Lumped Element Method) as an approximation of Finite Element scheme [119]. As the identification of masses is usually not problematic, the remaining parameters are the springs' stiffnesses. The MSS used in their case was based on a rectangular 2D mesh, therefore the two parameters based on the Lamé constants λ and μ are the edge (k_e) and face diagonal (k_f) spring stiffnesses. However it was not possible to obtain similar formulas for other mesh types, so the authors used a FEM based approximation. To obtain final values of stiffnesses the difference between the stiffness matrix of MSS and FEM was minimized to obtain the best results. This method fall into both categories: data-driven parametrization and solution based on analytical derivation.

   In Fig. 3.11 the original position of the sample is depicted in grey dotted lines, the reference FEM model in blue and MSS in red. Even though the maximal and root mean square error of this experiment were relatively equal to 5.50% and 3.36%, this method's drawback is definitely the fact, that a stiffness formula cannot be derived for any other type of mesh.

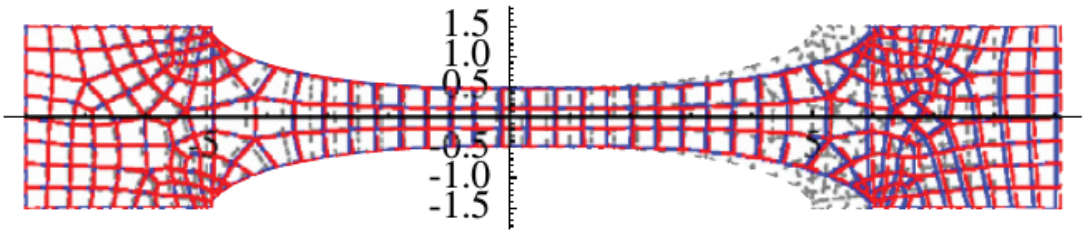


FIGURE 3.11: Linear elastic material simulated with hexahedral mesh [119].

The presented methods for stiffness derivation are focusing on a very narrow range of material properties. Vast majority aims at a correct simulation of linear elastic materials, which is the most important property of a soft tissue. However, the plausibility of their experiments usually is limited to visual effects and not numerical study. In the few cases, where the stiffness parameters are obtained using physical laws and/or FEM parametrization [99, 12, 119] we can observe major improvements of MSS, which show

that it is indeed possible to allow a physically-based MSS simulation under specific circumstances.

3.5 MSS Constraints

Among different types of constraints there are those, which control the length of springs, the surface area of faces or the volume of 3D elements [51, 52] or even anisotropy [24]. Simulating soft tissues seems to attract the topic of volume conservation or correct volume behavior, since this is one of the most important properties of incompressible bodies used in surgery simulation like internal organs.

3.5.1 Volume change

Bourguignon et al. proposed an approach based on tetrahedral and hexahedral meshes, to ensure volume conservation of an element (with a maximal relative error 1.5%) [24]. The volume-correcting force formulation uses the information from springs (stiffness, damping) and vertices (positions, velocities) to tie the vertices to the barycenter of a volume instead of treating the vertex' forces only along the 3 main dimensions. However, since the hexahedral meshes are not homogeneous in their approach (the masses are not distributed evenly) the model can be used only for limited types of bodies. Also, the authors don't associate the volume-correcting forces to the actual volume of mesh elements.

Mollemans et al. decided to deal with the volume conservation as additional term in the computation of forces [112]. The total force $\mathbf{F}_i$ on a node is composed of three elements: internal, external and volumetric components $\mathbf{F}_i = \mathbf{F}_i^{int} + \mathbf{F}_i^{ext} + \alpha\mathbf{F}_i^{vol}$. The real α coefficient is responsible for the impact of the force term and it is based on the influence of the elastic deformation and change of volume. The volume force itself is then dependent on the change of volume of every tetrahedral element j of a barycenter at point $\mathbf{x}_{b_j} = \frac{1}{4}\sum_{j=0}^{4}\mathbf{x}_j$ and it reads:

$$\mathbf{F}_i^{vol} = \sum_j (V_j - V_j^0)\frac{\mathbf{x}_i - \mathbf{x}_{b_j}}{|\mathbf{x}_i - \mathbf{x}_{b_j}|}. \tag{3.22}$$

As we can see, the corrective force is based on the change of the volume of a basic element, but in the general calculations only the biggest forces are taken into account under condition $\mathbf{F} > 0.1\ \mathbf{F}_{max}$. Such treatment ensures a good speed-up, but involves a major drawback for the accuracy.

To improve the accuracy, the approach of Baudet et al. incorporated the bio-mechanical properties of the material into the formulation of the spring stiffness [12]. The authors proposed a formulation based on the Young modulus and the Poisson ratio with force corrective terms to ensure the correct use of Poisson's ratio, and [119] a formulation based on the Lamé constants. Moreover, Baudet [12] introduced the stiffness formulations, which were discussed in Ch. 3.4.3 based on tensile and shearing deformations, but correct only for $\nu_{2D} = 1/3$ and $\nu_{3D} = 1/4$. They were complemented with an idea of correction forces to fix 2D or 3D MSS's error when different values of Poisson's ratio are used, but unfortunately they did not work with co-rotational modifications and were based on an area-preserved energy.

Jarousse et al. presented a simple method for volume conservation of myocardial non-linear, anisotropic elasto-mechanical tissues [79]. A hydrostatic work term (involving deformation tensor) is added to the energy density function, used later on to calculate the

volume preservation forces. But such a method destabilizes the system, therefore the model is not suitable for real-time simulation.

Kot et al. presented work on the adaptability of Poisson's ratio in simulation of isotropic homogenous elastic solids [92, 91]. To solve the problem of correct volume behavior, Kot et al. introduced the second degree of freedom into their MSS. The authors can use any Poisson's ratio and they proposed an extended MSS (viewed as a network of grouped neighbors) with dispersive momentum flow as the source of isotropic behavior of the material. When two nodes get close to each other, they not only repel each other but also start radiating momentum in random directions. The authors introduced then force, being a fraction of the momentum κ_μ, as an additional component of the regular forces. This phenomenon is introduced in the system as an additional force terms $\mathbf{F}_\mu$:

$$\mathbf{F}_\mu = -\kappa_\mu \Delta l, \tag{3.23}$$

and $\mathbf{F}_{diss}$, which is the term distributed uniformly over the other nodes and it is multiplied by a coefficient q, which is the ratio between the dissipative and direct momentum flow:

$$\mathbf{F}_{diss} = -q\kappa_\mu \Delta l. \tag{3.24}$$

Using this method, it is possible to operate with different kinds of volume conservation methods - global and local, with or without distance constraints. The method was introduced in both two and three dimensions. However, the run time was doubled in comparison to traditional MSS.

Another work in the correct volume behavior topic combined with new stiffness formulation was presented by Arnab et al. and it was based on the bulk elasticity [4]. As a continuation of the previously presented approach of the stiffness computation in point 4 of Tab. 3.1 in Ch. 3.4.3 on a surface mesh, we want to present their highly valuable extension to MSS. The authors do not only aim at the incompressibility of the material, but actually introduce the correct volumetric behavior dependent on the Poisson ratio into their surface-mesh MSS. The bulk stiffness k_{b_i} at a node i is defined by

$$k_{b_i} = \frac{E}{3(1-2\nu)} \sum_{i \in j} \frac{V_j}{l_i^2}. \tag{3.25}$$

The volumetric force $\mathbf{F}_k$ connected with this bulk stiffness is dependent on the change of volume ΔV, normal to the surface $\mathbf{N}_i$ (see Fig. 3.7 for details) and a weight coefficient α, which controls the distribution of this force:

$$\mathbf{F}_k = k_{b_i} \Delta V \; \alpha \; \mathbf{N}_i. \tag{3.26}$$

The global deformation allows conservation of the volume, however the model itself does not possess internal volume at all. For that reason is not generic in terms of mesh structure and availability for different simulated materials like internal organs.

3.5.2 Other Constraints

The approach of Duan et al. [51, 52] is combining the speed of MSS with the constraints of PBD [16, 50]. The authors decided to use the position-based constraints to correct the behavior of MSS. They introduced four compensation constraints: over-compression, over-stretching, spring direction and volume correction. They proposed a so-called Position-Based Mass-Spring System algorithm with constraint functions of the following form:

$$
\begin{aligned}
C_{stretch}(\mathbf{p}_i, \mathbf{p}_j) &= (1+\gamma_s)\cdot l_0 - ||\mathbf{p}_i - \mathbf{p}_j|| \geq 0,\\
C_{compress}(\mathbf{p}_i, \mathbf{p}_j) &= ||\mathbf{p}_i - \mathbf{p}_j|| - (1-\gamma_s)\cdot l_0 \geq 0,\\
C_{dir}(\mathbf{p}_i, \mathbf{p}_j) &= \theta - acos\left(\frac{(\mathbf{p}_j - \mathbf{p}_i)(\mathbf{p}_j^0 - \mathbf{p}_i^0)}{|\mathbf{p}_j - \mathbf{p}_i|\,|\mathbf{p}_j^0 - \mathbf{p}_i^0|}\right) \geq 0,\\
C_{vol}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4) &= \frac{1}{6}((\mathbf{p}_2 - \mathbf{p}_1)\times(\mathbf{p}_3 - \mathbf{p}_1))\cdot(\mathbf{p}_4 - \mathbf{p}_1) - V_0.
\end{aligned}
\tag{3.27}
$$

In these equations we can see, that the over-stretching and over-compression compensations are dependent on the critical stretching rate γ_s, the spring direction constraint is dependent on the critical rotation angle θ and of course the volume constraint is dependent on the initial volume V_0. In this work, the springs were limited to 10% extension/compression to respect the natural behavior of materials. The results show mean error of 3.3% for small and 4.2% for non-linear deformations in comparison with FEM, proving model's usability for real organ simulation. Unfortunately, the method is not universal and the mesh-dependent deformation ratios need to be set up by the user to achieve best results.

The method of Bourguignon et al. [24] was also used by Fritz et al. who wanted to adjust the MSS to be able to simulate the hyper-elastic behavior of heart tissue [58]. The adaptation was done using analytical derivation of energy density function which describes myocardial mechanics. This technique limits the use of MSS to very specific type of body and behavior, but also shows that MSS adaptation may be done using continuum mechanics-based laws.

Zhao et al. [171] presented a higher order Lattice Spring Model (LSM) with elements of previously developed Distinct LSM (DLSM) [170]. The model is supposed to be able to simulate various Poisson's ratio values without violating the rotational invariance. The model's stiffness parameters are derived on the Cauchy basis combined with the hyper-elastic theory. The model is generally used to simulate elastic and dynamic failure problems. Very recent work of Zhao introduced a new method combining modern physics with solid dynamics to deal with the Poisson limitation [169]. It is done by introducing a model with another spacial dimension to the lattice where 3D solids are viewed as mappings of hyper-membranes in 4D [169]. The model is used to simulate hyper-elastic solids and features numerical examples of elastic problems or dynamic motion of a soft robot. Several 4D lattice experiments were compared with a non-linear FEM and the experimental results have shown very good alignment with this method.

All these constraints aim at improving MSS with additional structures or force terms in different areas. Indeed it is the volume *conservation* which cut the attention of scientists, however we can clearly see, that many of them also put up with the problem of *correct* volume behavior by introducing possibility of using the Poisson ratio as a volume controlling coefficient.

3.6 Non-linear Behavior

Mass-Spring System is known for its simplicity and indeed this is the most valuable feature of the model. As we already discussed the basic MSS working scheme, there is no surprise that many researchers want to use it not only for linear (small) deformations, but also to model non-linearity of soft tissues. Non-linear behavior of soft tissues is indeed a topic, which requires our attention, especially in the field of biomedical simulation. Deformable bodies like real organs deform in a non-linear manner when big loads are applied to them and it is crucial in the applications like surgery simulation to be able to represent this type of behavior correctly. It can be understood that in the soft tissue simulation the non-linearity of this type of body is caused by its visco-elastic or elastic properties [88]. The non-linearity can generally be observed on stress-strain or stress-time curves, which is depicted in Fig. 3.12.

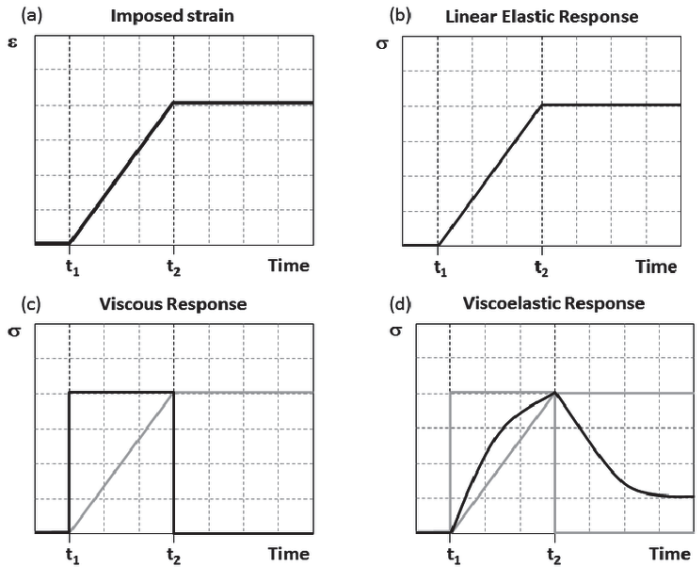


FIGURE 3.12: A stress-time relationship for different types of materials at loading between t_1 and t_2 and a static strain at $t > t_2$ [136].

Nevertheless, we can find different methods of introducing the non-linearity into MSS simulation.

Natural MSS Non-linearity

Lloyd et al. claim, that MSS have non-linear behavior (seemingly alike Green-Lagrange strain), which is visible in their experiments on a model derived using a linear FEM [99]. Miller et al. presented their novel method called Total Lagrangian Explicit Dynamic (TLED) algorithm, which is based on FEM [110]. San-Vincente in his thesis and article proposed a non-linear method for cubical MSS [156, 141]. Similarly to Lloyd et al., San-Vincente claims the non-linearity being a part of MSS and improves the behavior of the model by searching for valid stiffnesses based on the reference behavior and stress-strain and compressibility curves.

Additional Forces

A method called HEML, which stands for Hyper-Elastic Mass Links, is based on the non-linear visco-hyperelastic constitutive law and was introduced by Goulette et al. [66]. Their method use the additional force term to add non-linear behavior. Cui et al. presented a method based on the Duffing equation to model non-linear force in a triangular surface mesh used with haptic devices [38].

Non-linear Stiffness Formulations

Teschner et al. used non-linear stiffness in his multi-layer model to simulate craniofacial surgery [152] and he was followed by Chen et al., who also used non-linear stiffness formulation, but this time in a 3D triangular surface mesh [30]. Non-linear springs were also used by Basafa et al. thanks to the use of a novel force formulation [7].

Non-linear Parametrization

Da Silva et al. used a parametrization technique to find optimal masses, stiffness and damping coefficients to model non-linear oscillations [144].

Constraints

Duan et al. presented the position-based constraints (Eq. (3.27)) which introduce non-linearity in their position-based MSS [52]. Another idea for non-linear elastic simulation of facial tissue was introduced and used by Wang et al. in the facial simulation [157]. They based it on a dedicated elastic function, which takes a spring stiffness, change of spring's length and a non-linearity factor α as components.

So far the existing methods allow only manual adjustments or are specifically designed to model only particular type of tissue, therefore the non-linearity of deformable bodies is still an open topic.

3.7 Topological modifications

The topological modifications are invaluable part of surgery simulation. Indeed, in virtual surgery it is crucial to be able to cut or pierce the tissue, therefore models with possibility of real-time cutting are much more valuable as virtual reality tools than those which need some pre-computations after applying topological changes. These modifications may include tearing, refinement and de-refinement techniques, but here we focus on the cutting and piercing in particular. Note, that the most popular meshes when it comes to topological modifications are triangular and tetrahedral - these elements are simplest to subdivide into more triangular shaped elements, therefore possible refinement techniques are easily applied in comparison to *e.g.* hexahedral meshes, which pose a huge problem of T-junctions.

Paloc et al. presented a multi-resolution MSS working on tetrahedral meshes [134]. The method is able to handle haptic feedback and uses change of the resolution dependent on the distance from the body. Authors use dynamic update of the mass, stiffness and damping coefficients to simulate a MSS with high visual resolution.

Starting the topological cutting topic from an article of Cotin et al. from 2000, who introduced one of the first models for surgery simulation including topological modifications and even force feedback [33], we can already see how long time ago the need for such a simulation was recognized. The authors used computation and constant update of local tensors to know when to remove a tetrahedron at which the cutting takes place. However, this method requires the tetrahedra at the cutting place to be relatively small and it is possible that more than only the cut element will be removed from the mesh. Additionally to cutting, the local tensors and stresses are computed to verify if the tissue is not torn - if they exceed a given limit - tetrahedra are automatically removed.

Bielser et al. presented a cutting technique based on a pre-defined cutting scenarios for triangles, which was extended to tetrahedra [21]. Such a method limits the number of possibilities when it comes to cutting, but the method allows fairly fast recognition of the cut patterns and does not allow unexpected instabilities to tamper with the model.

Ganovelli et al. used a Multiresolution Triangulation (exceptionaly in this section referred to as MT) method for decomposition of their tetrahedral mesh [63]. The mesh is generally composed of segments with particular borders, so an update of a segment with use of MT will exchange such a segment with a new one (refined or simplified, according to the need), but with the same border. After discovering a collision of the mesh with a scalpel, the intersecting segment is modified to not have any intersections with the tool and then the MT algorithm is applied to update the mesh.

Flechon et al. used combinatorial maps and LCC+MSS layer (see Ch. 3.8.2 for details) to facilitate all the topological operations including cutting and adaptive refinement [56]. The method works well during an ongoing simulation while allowing operations like cutting in the same time.

As we can see, the topological modifications are handled well in MSS in the literature. It is another advantage of using MSS for surgical simulation in virtual environments, where user is able to modify the tissue according to the needs.

3.8 Presentation of Our Framework (TopoSim) Suitable for MSS

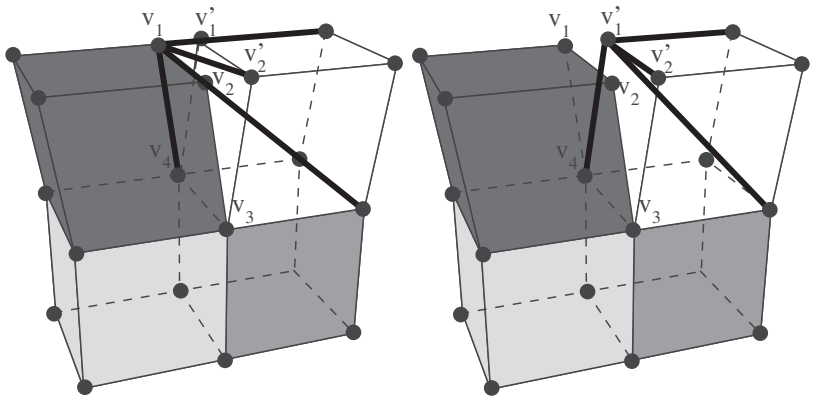
In this section we introduce the topological structure of our MSS as well as the underlying LCC+MSS layer. It is followed by the implementation of our framework called TopoSim. The framework is used to integrate our MSS and the methods we developed and present them in the following chapters. The objective of TopoSim framework is to be as generic as possible. This means that it allows an easy use of different simulation techniques on various models. In this context we are most interested in developing the Mass-Spring System and introducing various mesh types.

3.8.1 The Topological Data Structure

Graph

Different authors decided to use different underlying structures to help to deal with the meshes and the manner in which they store information. The simplest structure used for this purpose is a *graph*. Undirected graphs are used to structure simple and usually small mass-spring systems. In such a case the vertices are the nodes of a graph and the springs are its edges. The structure can hold necessary information about the physical parameters of the mass-points and springs as well as a list of all the connections to other nodes. Such a structure may be useful in case of small meshes, but its simplicity becomes a huge drawback when dealing with more geometrically complex models. Especially the topological modifications pose a computationally expensive task, where the lists of neighbors must be iteratively updated depending on the applied changes.

For example, to cut an object between two adjacent hexahedra (*i.e.* two hexahedra sharing a face), particles of the face would need to be duplicated (see Fig. 3.13). To detect if a particle needs to be duplicated or not, we need to search for existence of a path of adjacent elements joining the two cut hexahedra. This is not possible directly in a graph. Thus, a data-structure has to be added to represent elements by storing indices of its particles and indices of its adjacent elements. Then, we also need to add to each particle the indices of its incident elements. These additional data-structures and links need to be initialized and updated coherently after each operation. Moreover, other data-structures and links are required to implement other topological modifications.



(a) Cutting before update (b) Cutting after update

FIGURE 3.13: Cutting with necessary update of the spring extremities. (a) The bold segments represent the inner springs wrongly attached to particle v_1 after the cutting. (b) The same springs updated after a post-processing step.

Combinatorial Map

A better structure than the aforementioned graphs, which can be used to describe the 3D mesh topology is called **combinatorial map** (3-map) [95, 40]. Let us consider the example of Fig. 3.14. 3-maps consist of darts, which are the directed arrows interconnected by β links. We can get the information about a dart from β links in the following way:

- β_1 - the link connecting a dart with the next dart of the same face;
- β_2 - the link connecting two darts, which belong to the same edge, but different faces;
- β_3 - in 3D it is the link between two darts of the same edge and face, but different volumes.

β_0 is a notation for β_1^{-1} *i.e.* reverse dart of β_1. 3-map can retrieve information about cells from the beta links. For example dart 2 can be retrieved as $2 = \beta_1(1) = \beta_0(3)$ and dart 9 can be retrieved as $9 = \beta_1(11) = \beta_0(10) = \beta_2(6)$.

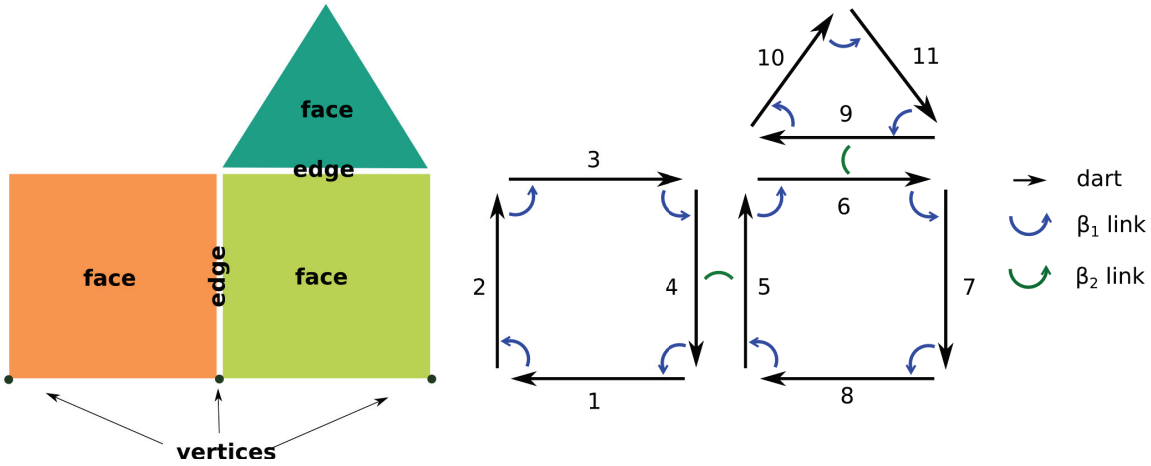


FIGURE 3.14: 2-map: space subdivision (left) and c-map structure consisting of darts and β_i links (right).

For an n-dimensional combinatorial map, with dart $b \in B$, where $i \in [0; n]$ [41] the following statements are satisfied:

1. Dart b is $i - free$ when $\beta_i(b) = \emptyset$.
2. For $i \in [2; n]$ each β_i link is its own inverse, meaning that if dart b is not $i - free$ then $\beta_i(\beta_i(b)) = b$. This rule does not hold for the β_0 and β_1 links, as they are each other inverse: for two darts b_0 and b_1 such that $\beta_1(b_0) = b_1$ we have $\beta_0(b_1) = b_0$.
3. If for an existing dart b_0 there is no dart such that $\beta_1(b_1) = b_0$, then $\beta_0(b_0) = \emptyset$.

If for a combinatorial map there are no $i - free$ darts, the maps is said to be without $i - boundary$; Adequately, if an n-map is without $i - boundary$ in all n dimensions, the map is said to be without $boundary$.

Another important notion describes the incident and adjacency relations. If two i-cells share a common $(i - 1)$-cell, they are adjacent (for example two volumes sharing a common face). If one cell belongs to the boundary of another cell, these two cells are incident (for example a vertex belonging to the boundary of a face).

The main interest of using a topological model like combinatorial map is that we are sure that all the topological relations are described and can be retrieved directly. Existing topological constraints ensure the topological validity of the described objects and can be used to test the consistency of implemented operations. Moreover, we can use many existing topological operations, which are proven to work well, that can serve as basic tool to implement high level operations in our physical simulation.

A topological description based on a 3-map is the basis of the model. Other information, like all the aforementioned positions, masses, etc., must be added to the model to actually create a Mass-Spring System ready for simulation. The information are added to the particular cell types according to their physical properties. This is the next important layer of our MSS.

3.8.2 The 3D LCC+MSS Model

Flechon added information to a 3-map to construct the LCC+MSS model which is suitable for a MSS based on the Baudet formulation for hexahedral mesh [12]. This formulation is available for hexahedra including four inner-diagonal springs.

Physical Parameters and Topology

At this point we were using two types of MSS springs: edge and inner diagonal, with the formulation of the stiffness parameters as in the work of Baudet et al. [12] (Eq. (3.28)). We did not however use their force correction terms because of the limitations they pose when rotational deformatons are applied to a model. This aspect of the initially used MSS is the most important in our further developments, so here we present it in depth.

In pursuit of physically realistic deformations, Baudet et al. decided to introduce a new MSS model which finally deals with the problem of controlling the physical behavior of simulated deformable solids in a satisfying way. The key to their solution was to include the correct mechanical properties of an elastic body (namely Young's modulus and Poisson's ratio) into the MSS itself. They presented a parametrization in 2D and 3D and we focus on the latter one, since our interest lies in the simulation of deformable solids.

The parametrization approach starts from identification of two deformations: uniaxial tensile experiment and a shearing stress. The choice of these two types of deformations was dictated by the fact that the simulated tissue is elastic (therefore a tensile test is crucial) and has to correctly represent the shearing modulus G (thus a shear stress). First, the authors assumed three possible configurations of the springs in a simplest 3D element (Fig. 3.15 a): the configuration (b) with only face-diagonal springs; configuration (c) with only inner-diagonal springs; configuration (d) with both types of springs combined.

Each of the configurations was evaluated in terms of numbers of equations needed to solve a system based on the Lagrangian framework for tensile and shearing deformations consisting of three types of unknown stiffness coefficients: k_e for the edge springs, k_f for the face-diagonal springs and k_c for the inner diagonal springs. Of course, to solve the system, the number of equations must be equal to the number of unknowns. If the number of equations exceeds the number of unknowns (resp. if the number of unknowns exceeds the number of equations) the system is said to be over-constrained (resp. under-constrained). After calculations the authors found out that, taking into account the two different deformations, all of the configurations result in an over-constrained systems. It turned out that among the three possibilities, the configuration III is *the least* over-constrained of them

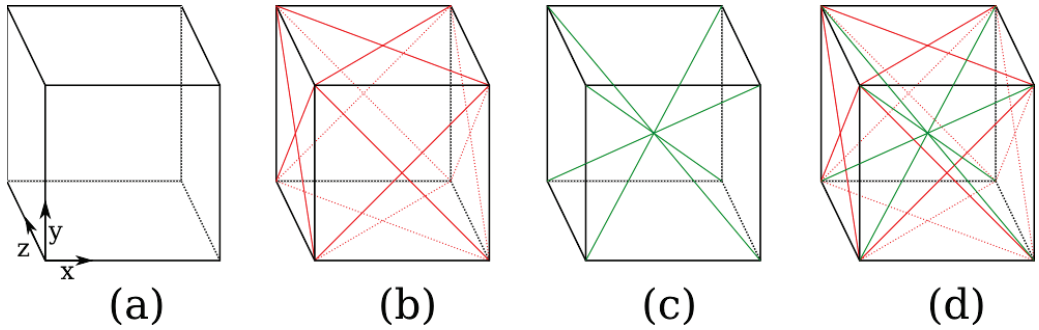


FIGURE 3.15: Three possible configuration of springs in a cubical element: (a) face-diagonal springs, (b) inner-diagonal springs, (c) face-diagonal and (d) inner-diagonal springs.

all. In the same time the inner-diagonal springs are enough to reproduce the thinning, which occurs at the body elongation in the tensile experiment, so the choice of spring configuration III is well justified.

From the derived systems of equations the authors found the stiffness coefficients obtaining as follows:

$$\begin{aligned} k_e &= \frac{El_0(4\nu+1)}{8(1+\nu)}, \\ k_c &= \frac{3El_0}{8(1+\nu)}, \\ k_f &= 0. \end{aligned} \tag{3.28}$$

Data Structures

In the LCC+MSS model there are two following data structures:

- `Particle` - associated to each 0-cell thanks to a 0-attribute; the structure stores information related to a particle: its mass, velocity, acceleration and the sum of the forces applied to it. Moreover, it stores the list of the inner springs attached to it.
- `Spring` - associated to each 1-cell thanks to a 1-attribute; this structure stores the physical properties of a spring: its initial length, its stiffness and its two extremities (thanks to two pointers to the two `Particle` connected by the considered spring).

Contrary to the described particles and springs the inner-diagonal springs are not associated with any cells of the 3-map, but directly associated to the object.

Fig. 3.16 represents two adjacent cubes described by a 3-map (a) and the corresponding LCC+MSS structure (b). Darts are displayed as arrows.

Improvement of the LCC+MSS Data Structure

To be able to simulate the topological operations faster and more efficiently we decided to improve the initial 3D LCC+MSS model. The improvement makes it more generic and allows easier implementation of new simulation methods. The main idea is to associate the information to the model's elements, depending on the underlying layers. The simplification we apply involves a generic simulation loop and an alteration to the structure of the model allowing minimal effort and computation time when applying topological modifications. The proposed improvement involves changed architecture of the LCC+MSS

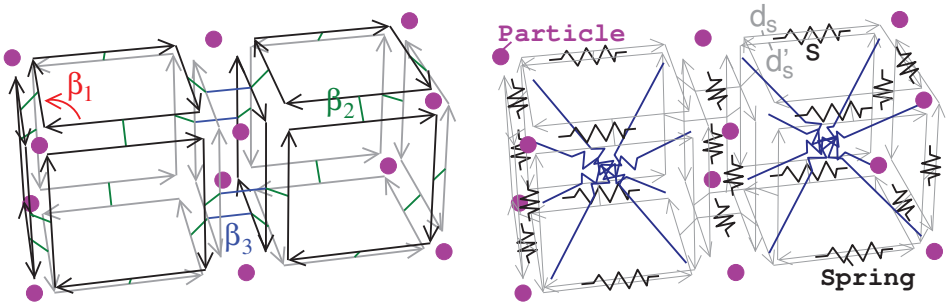


FIGURE 3.16: (a) Two adjacent cubes described by a 3-map. Each cube has 6×4 darts. (b) LCC+MSS underlying structure of the same cubes: particles and springs associated to i-cells and inner particles associated to an object.

model to solve the problem of redundant operations when we cut an object. In fact the duplication of particles and springs and re-assignment of the correct springs to appropriate particles weakens computational efficiency of the simulation.

At this time, we consider only hexahedral elements due to the physical models used, but the structure is suitable for any topology. The model follows two newly introduced principles:

1. Storing the darts of each element in 3-cells of the 3-map.

As the first principle we add an array called `dart` to each 3D element. The array contains all the darts of the element grouped by faces *i.e.* association of a `dart[6][4]` array with each hexahedron. The initialization of these arrays is performed at the construction of the 3-map using the convention given in Fig. 3.17. With this structure, all points of a given hexahedron can be retrieved thanks to the stored darts. For example, point X_0 in Fig. 3.17 is obtained from `dart[0][3]` and point X_1 is obtained from `dart[4][2]`. Note that each point of an hexahedron can be retrieved by three different darts. For example X_0 can also be retrieved thanks to `dart[1][0]` and `dart[4][1]`.

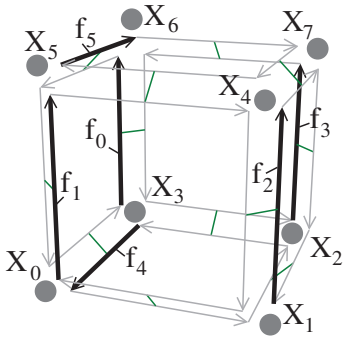


FIGURE 3.17: Convention used to store darts of a hexahedron composed of points X_i with $i \in \{0,\ldots,7\}$. f_j = `dart[j][0]` with $j \in \{0,\ldots,5\}$ corresponding to first darts of each face (the bolded arrows).

One major interest of our proposal is to provide a direct access to all the incidence relations between cells, allowing a direct access in $O(1)$ to physical and topological information required by different physical models. This direct access is done either by using the relations given by the 3-map or by the relations given by the `dart` array.

2. Associating each physical information directly with the corresponding cells of the 3-map.

The second principle is based on the fact that the physical information associated to the i-cells (with $i \in \{0, 1, 2, 3\}$) of the 3-map is depending on the physical model chosen. Consequently, the direct association of the physical information to the corresponding cells allows to consider easily different physical models. In the next subsection, we present the use of our proposal for a mass-spring system, but any physical model can be envisaged by associating the i-cells to i-attributes with the appropriate physical information or force calculation.

As we explained in the initial definition of the LCC+MSS, the topological-physical model [55] associates physical entities to cells (similarly to our new solution), except for inner springs, which are not associated with elements. Moreover, each spring stores its two extremity particles to be able to accumulate the force of this spring to the particles.

With the previous solution extremity particles of springs may have to be updated after a cut. For example, in Fig. 3.13, the cut of the two top hexahedra leads to the split of vertices v_1 (resp. v_2) into two particles v_1 and v_1' (resp. v_2, v_2'). However, after this cutting operation springs of the top-right hexahedra are still associated with original particles, so v_1 and v_2 instead of v_1' and v_2' (see Fig. 3.13a). To solve this problem a post-processing is applied after the cut to update the incorrect extremities of springs (see Fig. 3.13b).

The solution proposed by Flechon is to update all the springs having as extremity the particles involved in the cutting. This requires to store in each particle the list of the springs attached to it. Then, when a particle is duplicated due to a cutting, an iteration is performed through this list to update the extremities of the springs to point to the new particles. Moreover, the list of springs is split in two parts: the first part for springs incident to v_1 and the second one for springs incident to v_1' (the new particle). This treatment requires a complex and time consuming processing.

With the new improvement, as the darts of an element are constant whatever the topological modification applied on its neighboring elements, no change is required for cutting. For example, the darts of the two hexahedra split in Fig. 3.13 are still valid after the cutting. Moreover, as springs and inner springs' extremities are only defined by using the darts of the hexahedra, they are still valid after the cutting operation without any updating. Thus, our algorithms with topological modifications are simpler and the method is faster than Flechon's.

Computation of the Force

To compute the forces generated by springs (supported both by 1-cells and 3-cells), each spring needs to know its two extremity particles in order to retrieve their position and velocity. This can be retrieved thanks to the 3-map and to the `dart` array, without storing explicitly the link between springs and particles.

- For each spring s associated to a 1-cell, one dart d_s of the edge is directly known thanks to the 3-map (see Fig. 3.16 (b)). The first particle extremity of the spring is given by 0-attribute (d_s). The second particle extremity of the spring is given by 0-attribute (d_s') with $d_s' =$ `other_extremity(`d_s`)` the dart associated with the second extremity of the edge.

- For any inner spring s_i associated to a 3-cell, its two particles extremity are directly retrieved by the `dart` array. For example for s_0, its first extremity is 0-attribute (`dart[1][0]`) and its second extremity is 0-attribute (`dart[3][1]`).

Thanks to this new way to associate physical information to a 3-map, the forces applied on the object are computed in a generic way by performing a loop on all the relevant i-cells, that is the i-cells affected by the force computation (*e.g.* the 1-cells and 3-cells for MSS). Each physical information must define an appropriate `add_force()` function that computes the forces resulting from this i-cell.

In practice, it means that we need to iterate through all the enabled i-attributes (*i.e.* the attributes which are associated to cells of the 3-map). The genericity of this approach lays in the force computation method, which is the same for all the applicable cells. It is important to note that in a general simulation loop the call for calculation of forces is always done in the same way for any integration scheme. The same principle stands also for other than MSS simulation methods if they are incorporated in our framework. It is not dependent on the MSS structure itself, therefore it can be used by any available technique.

3.8.3 Structure of Our Model in TopoSim Framework

In this section we introduce the general structure of our framework.

The simplified structure of our software, called TopoSim, is depicted on Fig. 3.18 as a UML class diagram.

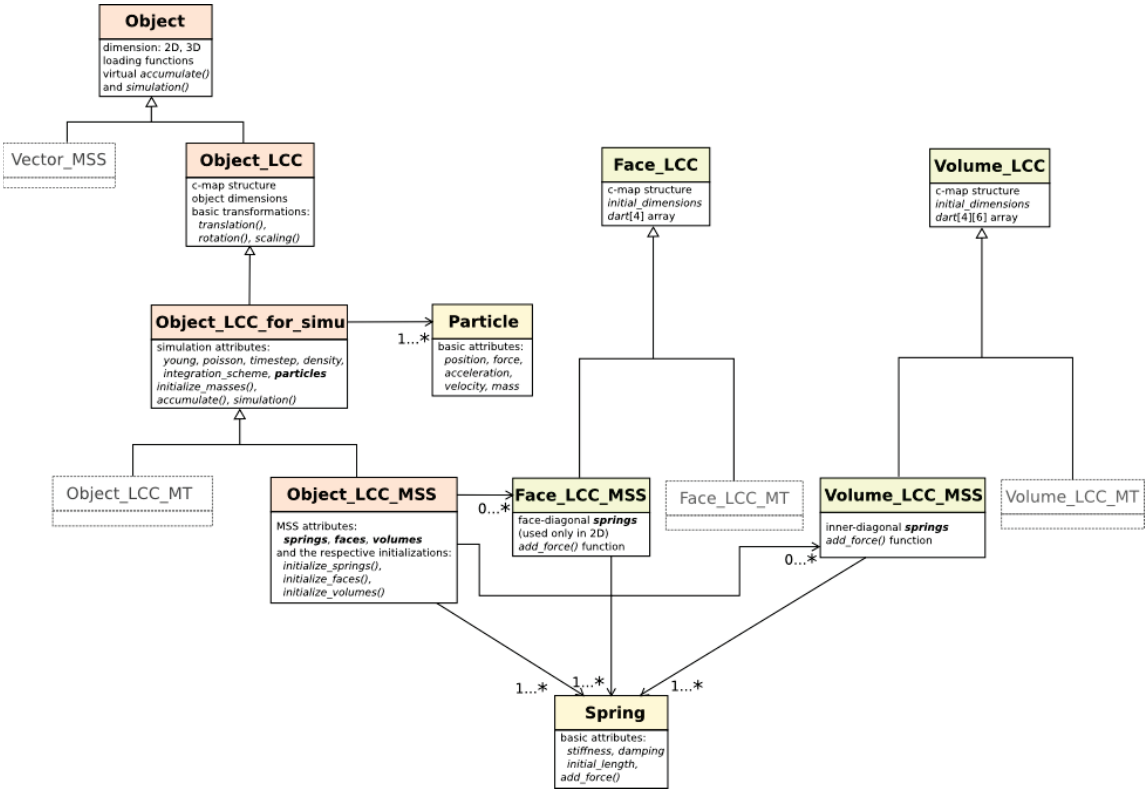


FIGURE 3.18: UML class diagram of TopoSim basic simulation components.

- A basic class, which is crucial for our MSS simulation environment is `Particle`. Its basic attributes are particle's position, acting forces, its acceleration, velocity and mass.

- Similarly to the `Particle` class, the `Spring` class holds basic information about each spring, like its stiffness, damping and initial length, as well as a crucial *add_force()* function, which computes the forces acting on the particles due to the spring elongation and accumulates them adequately. It is in this function, that we can specify the force formulation used by the simulator.

- The main class `Object` is responsible for loading and defining 2 and 3-dimensional objects. Having defined this class we can already implement new data structures. One of them may be `Vector_MSS`, which is a simplified MSS based on a graph structure. Its particles are stored in a vector and the model is a basic data structure without use of topology.

- Looking further at the UML diagram we have the layer of `Object_LCC` class, which belongs to `Object` and which is responsible for building the LCC structure (Ch. 3.8.2); *i.e.* the 2-map or 3-map, holding object dimensions and providing basic Euclidean transformation functions, like translation and rotation but also scaling. It allows to introduce the geometry and topology to our object.

- To create the simulation environment we have the class `Object_LCC_for_simu`, which inherits from `Object_LCC`. Additionally, this class holds the information about the simulation, that is the time step, integration scheme and physical body properties (values of Young's modulus, Poisson's ratio, body density). It is also responsible for initialization of the `Particles` and their masses using associated `Particle` class.

- Having the proper LCC simulation environment we can introduce the more detailed simulation method of the next layer. One of them may be Mass-Tensor model (implemented in the corresponding class `Object_LCC_MT`), which is based on a simplified FEM approach (Ch. 2.3.1). In our case the model is of course the MSS, represented by the class `Object_LCC_MSS`. This class is responsible for the treatment of `Springs`, faces (elements of `Face_LCC_MSS` class) in 2D, and volumes (elements of `Volume_LCC_MSS` class) in 3D. For that we need the corresponding initialization functions: *initialize_springs()*, *initialize_faces()* and *initialize_volumes()*. To serve its purpose correctly, the `Object_LCC_MSS` is associated to the aforementioned classes and allows access to their attributes during a simulation.

- The other two classes in the same layer, associated directly with the `Object_LCC_MSS`, are `Face_LCC_MSS` and `Volume_LCC_MSS`. Both of them inherit from the corresponding `Face_LCC` and `Volume_LCC` classes, which hold the basic 2-map or 3-map structures of those elements, their initial dimensions and the *dart* array. These two classes can be used also by the MT method to determine the corresponding `Face_LCC_MT` and `Volume_LCC_MT` classes. The crucial information and functionality of the `Face_LCC_MSS` class is the association with face-diagonal springs and the *add_force()* function, which accumulates the forces on those springs. Similarly, the `Volume_LCC_MSS` class introduces the inner-diagonal springs for each volume and the corresponding *add_force()* function.

To perform a simulation we need two main functions, which we did not mention before: *accumulate()* and *simulation()*. Let us study the following simulation loop algorithm:

Both of them are defined in the `Object` class as virtual function, to then be specified in the proper classes of the simulation methods, which belong to `Object` in the first place.

Algorithm 1: TopoSim simulation loop algorithm

forall *Objects* **do**
- `Accumulate`
  - **forall** *Particles* j **do**
    - accumulate external forces: $\mathbf{F}_j \leftarrow \mathbf{F}_{ext}$;
  - **forall** *Springs* i **do**
    - compute forces $\mathbf{F}_i$;
    - accumulate forces on the extremity particles:
    - $\mathbf{F}_{p1} \leftarrow +\mathbf{F}_i$;
    - $\mathbf{F}_{p2} \leftarrow -\mathbf{F}_i$;
- `Simulation`
  - **forall** *Particles* j **do**
    - compute the new positions $\mathbf{x}_j$ using an integration scheme given $\mathbf{F}_j$

Thus, in the case of MSS method to run the simulation we need to first call the *accumulate()* function on all the existing objects of `Object_LCC_for_simu` class. This function iterates through all the existing springs and computes and accumulates the forces on all the particles belonging to those springs. Then, the second step is the *simulation()* function (also run on all the `Object_LCC_for_simu` objects) which is responsible for updating the positions of all the particles using specified integration scheme and depending on the previously accumulated forces. Both functions are parallelized using Intel® TBB library algorithms, which largely improves the performance of the simulations. The function *accumulate()* is parallelized on particles and then springs and adequately, the function *simulate()* is parallelized on the particles.

Thanks to this particular architecture we are able to easily add and adjust the simulation methods, chose among different integration schemes and introduce new functionalities. The approach we chose is generic and simple to understand, which facilitates the workflow for any user and developer. In this section we have explained the topological structure and implementation of our framework in detail. It allows easy implementation of new solutions, straight-forward adjustment of simulation methods to the topological structure and generic force accumulation loop. Naturally, the computation of the acceleration of the object and its integration in order to obtain the velocity and position is the same for all physical objects who are based on the Newtonian dynamics.

3.9 Conclusions

This chapter has presented the Mass Spring System in its fullest form. Knowing the capabilities and limitations of this simulation technique we are able to verify the most important contributions which need to be made to reach our goal: be able to simulate deformable bodies - real tissues - with speed and accuracy. We know that it is crucial to define and deal with the MSS instabilities and limitations: Cauchy's limitation, MSS wrinkles and buckling phenomenon. Researchers constantly work on improvements of this method seeing a potential laying in this simple yet capable simulation technique. Looking at the existing MSS methods one may draw a conclusion that an advantage of using FEM instead would be the possibility of applying different physical laws in form of constitutive functions and precise boundary conditions to the model. Nevertheless, this better description of the mechanical behavior in case of FEM is counterbalanced by the advantages of mass-spring models, namely the fast computation and the effortless topological modifications in real-time without additional pre-computation (which is still necessary when using FEM). Indeed, it is possible to simulate complex behavior with MSS without resigning from the real-time capabilities of this modeling technique.

We have also presented the topology characteristics used for MSS (and other) simulations. The additional MSS+LCC layer which improves boosts the MSS capabilities was described in detail thanks to the previous work of Fléchon et al. and introduction of our improvement. Additionally we have covered the topic of the initial implementation of our solution, which makes a generic basis for the contributions which are to be introduced.

In next chapters we present our methods to solve several MSS limitations and introduce correct global behavior. We will show that it is possible to be able to simulate elastic deformable solids in an adjustable manner, which does not require complex precomputations. In our work we focus on the correct MSS behavior and controllability of the system. In the same time we introduce physically-based methods for MSS simulations with use of constitutive law and continuum mechanics. Using appropriate physical parameters should be the main focus of a correct physically based simulation.

Chapter 4

Linear and Non-linear Behavior of Biological Soft Tissues

In this chapter we present a constitutive law which describes the behavior of real organs (like liver, kidney, spleen) and their deformations up to 100% strain. Our goal is to be able to represent the non-linear behavior of soft tissues with our simulation tool. For that purpose we present a non-linear formulation for computation of forces inspired by the constitutive law. We describe the method in which the constitutive law was obtained - a work of Nicolle et al. [124, 123] - and how we adjusted it to use in the MSS (presented in the previous chapter: Ch. 3). In the following sections we describe the experiments performed on real tissues, the resulting constitutive law and our non-linear MSS formulation. In order to replicate the behavior of the constitutive law by our improved MSS we present the results of experiments performed using our TopoSim framework. We finish the chapter by discussing the outcome, applications and possible improvements to the presented method.

4.1 The Experimentally Obtained Constitutive Law

The work of Nicolle et al. [124] describes a strain-hardening bi-power constitutive law for non-linear behavior of biological soft tissues. The general principle states that at large strains the strain hardening region of the stress-strain curve of such a tissue's deformation is placed between the yield point and the breaking point. The point of this constitutive law is to define the non-linear behavior of biological tissue with as little parameters as possible. Such a definition of the physical body behavior may be used for modeling purposes and it would allow more realistic body representation. We chose (among others) the Nicolle's experiments to validate the response of our MSS model. The force formulation presented in this chapter is inspired by the Nicolle's model to make the response of our MSS model also non-linear.

4.1.1 Real Tissues Experimental Setup

Nicolle et al. performed experiments on several types of internal porcine organs (kidney and liver) [124], which was also extended to spleen tissue [123] and pancreas [122]. The experiments were performed on a real tissue (extracted from 6-months old pigs), using a rheometer (see Fig. 4.1) with parallel plate configuration. Such a rotational rheometer guarantees experimental results where the edge effect does not appear.

The tissue was cut into circular slices of a 15 mm diameter and glued to the top and bottom plates of the device. The thickness varied from 2 to 3.7 mm. When the top plate

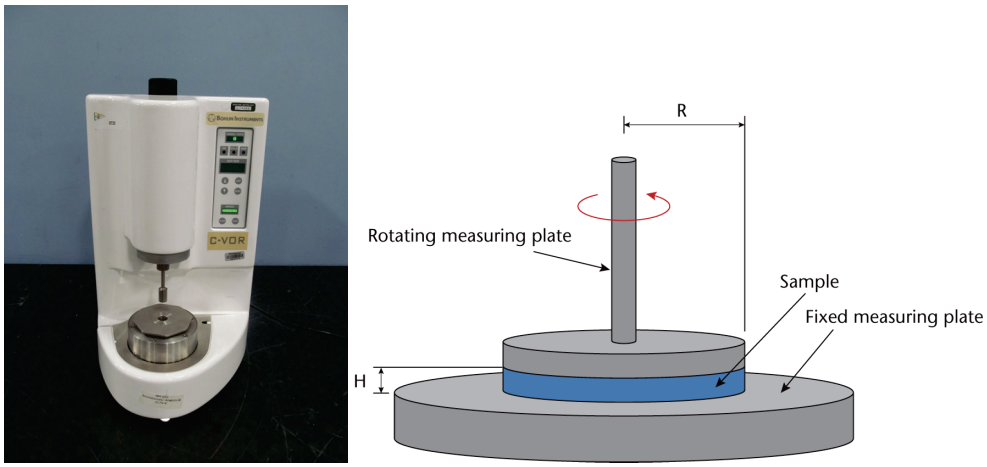


FIGURE 4.1: A Bohlin C-VOR rotational rheometer (left) with parallel plate configuration (right).

starts rotating, the device measures the torque acting within the sample on the top plate in response to the movement.

The main advantage of this type of test is that the torsion can be understood as a local shearing deformation (Fig. 4.2). Instead of the rotating top plate, one can imagine two parallel infinite plates with the sample in between. When the top plate of the rheometer starts moving we can observe local shear stress and extract the information about strain from the forces acting on the top plate within the sample.

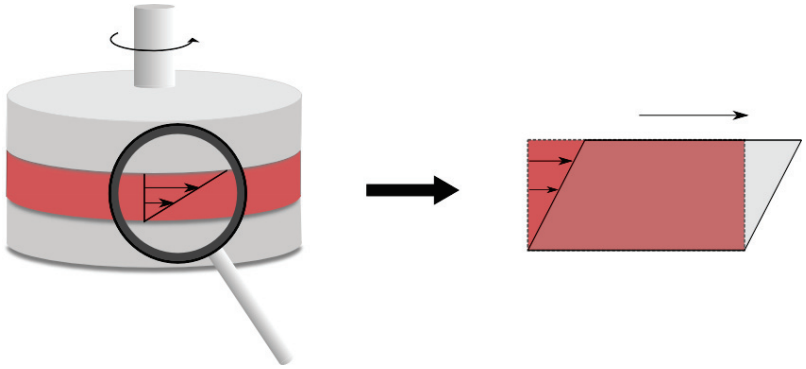


FIGURE 4.2: Torsional deformation (left) understood as local shearing (right).

Each tissue sample was used to perform two experiments: one with a small strain $\gamma \in \{0.1\%, 1\%\}$ and one with constant strain rates $d\gamma/dt$ being $0.0151\ \mathrm{s}^{-1}$, $0.133\ \mathrm{s}^{-1}$, $0.67\ \mathrm{s}^{-1}$. For our purposes, we are interested the most in the second part of the experiment to study the behavior of linear body deformations at different rates of the speed of deformation.

4.1.2 Strain-Hardening Bi-Power Law

The analytical model proposed by Nicolle describes the non-linear shear behavior by predicting the strain-hardening of the tissues like kidney, liver and spleen. The stress-strain relationship, where σ is the stress, $\gamma(t)$ is the strain at time t, has the following form:

$$\sigma[\gamma(t)] = \int_{-\infty}^{t} G(t-t', \gamma(t) - \gamma(t'))\ \dot{\gamma}(t')dt'. \tag{4.1}$$

The most important component of this relationship is correctly defining the relaxation modulus G. Here, the re-definition of this parameter includes dependency on the strain γ:

$$G(t,\gamma) = \frac{K}{\Gamma(1-n)} t^{-n} \left\{ 1 + \left(\frac{\gamma}{\gamma_{lim}} \right)^{p} \right\}, \tag{4.2}$$

where K is the consistency, n is a linear constitutive index, p is the hardening constitutive index, γ_{lim} is the linear visco-elastic limit and $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$ is the Gamma function. In case of small strains, so when $\gamma \ll \gamma_{lim}$, the relaxation modulus reduces to

$$G(t) = \frac{K}{\Gamma(1-n)} t^{-n}, \tag{4.3}$$

so it is dependent only on two parameters: K and n.

Experimentally obtained parameters for each type of tissue give us the corresponding values of the key parameters, which are presented in Table 4.1. Note, that the unit of the consistency $K = [\text{Pa s}^n]$.

Tissue	K	n	γ_{lim}	p
Spleen	758	0.1043	0.73	1.6
Liver	861	0.1138	0.40	4.17
Kidney	1818	0.0892	0.11	2.64

TABLE 4.1: The parameters of three types of internal organs obtained by fitting the curve from Eq. (4.1) to the experimental results.

Since the torque can be represented in terms of time, we present as an example the analytical stress of a liver tissue as a function of strain for the three different strain rates on Fig. 4.3. The red line marks the linear deformation limit, where the lines are still straight and we cannot yet observe the visco-elastic deformation.

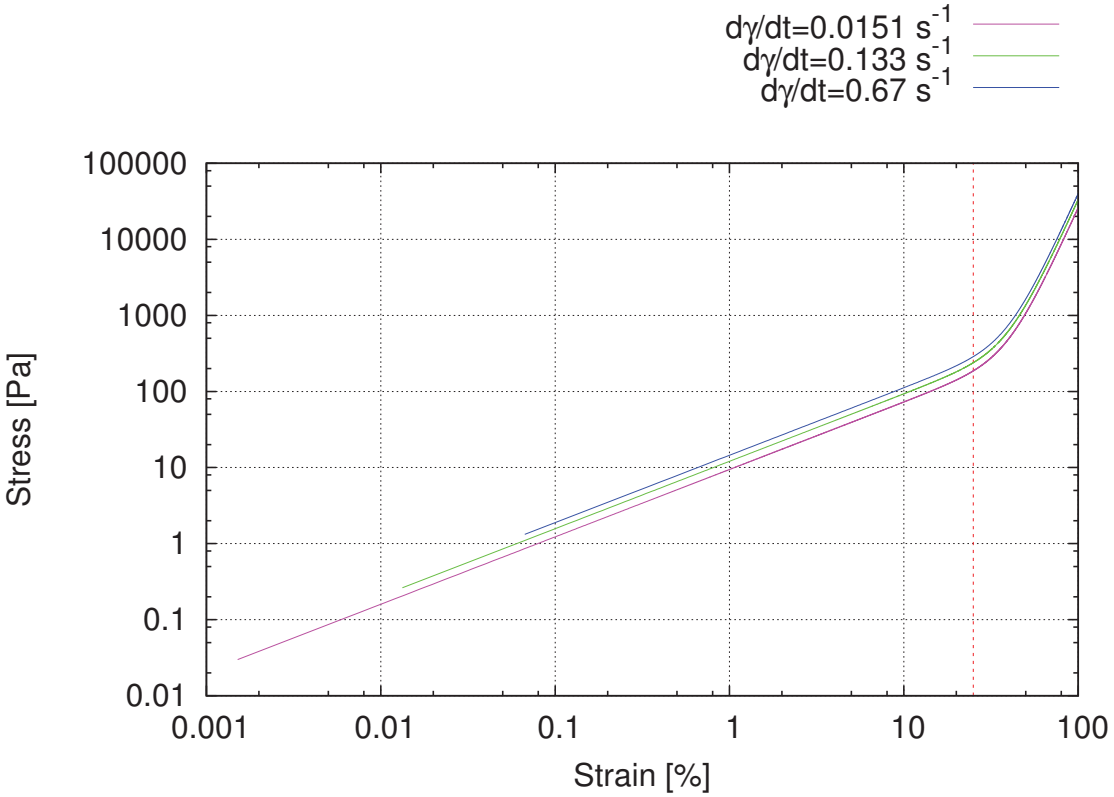


FIGURE 4.3: Analytical bi-power law of liver for three different strain rates $d\gamma/dt$ in log scale for big deformation in stress-strain.

4.2 Integration of the Constitutive Law in MSS

We have modified the classical spring force computation to describe the main features of the non-linear visco-elastic behavior of soft tissues. To be able to model non-linear behavior of soft tissues in MSS it is not enough to modify stiffness and/or damping coefficients. We need to modify the formulation of forces or formulate an additional force term to deal with this type of behavior. In our work we focus on a simplified model being able to incorporate the non-linearity of abdominal organs. The model assumes a *homogeneous*, *isotropic* material with *visco-elastic* properties of internal organs.

The new formulation of the force involves viscous and elastic terms derived from the constitutive law. It is important to include the elongation of the spring as well as the speed of the elongation (called elongation velocity). The classical formulation is then substituted by the following:

$$\begin{aligned} F_{nl} &= \left(d \cdot \frac{dl}{dt} + (k \cdot |l - l_0|^{1-n} \cdot \frac{l - l_0}{|l - l_0|}) \right) (1 + \alpha |l - l_0|^{p-(1-n)}) \\ \alpha &= \left(\frac{1}{\gamma_{lim} \cdot l_0} \right)^{p-(1-n)} . \end{aligned} \tag{4.4}$$

Naturally, d, k, l, l_0 parameters stand respectively for the damping coefficient, stiffness coefficient and current and initial length of the spring. In accordance with the previous section and the formulation of the bi-power law, γ_{lim} is a strain limit, n is a linear constitutive index, p is the hardening constitutive index. The term in first brackets of the Eq. (4.4) is responsible for the viscous behavior of the body.

Each of the parameters used in both the constitutive law and the new non-linear formulation of forces is responsible for a specific part of the deformation process. Fig. 4.3 presents the non-linear analytical stress-strain curve of the liver at $d\gamma/dt = \{0.0151, 0.133, 0.67\}$ s^{-1}. The matching curves, which we want to obtain using the new force formulation, will be shaped like the analytical ones thanks to three main equation parts matching three regions of the plot. The placement of the beginning of the strain hardening part of the deformation called strain limit is dependent on the value of γ_{lim}. It is marked as the red vertical line. Then, the first (linear deformation part) slope, on the left from the strain limit, is dependent on the value of the linear constitutive index n and more precisely on the power $1 - n$ from the first brackets in Eq. (4.4). Finally, the non-linear slope, being on the right from the strain limit, is dependent in the same way on the hardening constitutive index p and the power $p - (1 - n)$ in the second brackets of Eq. (4.4) and in α term.

This approach allows an easy way to implement solution for MSS to incorporate the non-linear behavior of real bodies. There are no additional structures needed in the MSS formulation and simply by changing the values of the key parameters, like p, n or γ_{lim} we can change the behavior of the simulated body in order to reproduce the real experiment. Once the parameters for a particular type of tissue are adjusted the simulation can be performed on any mesh. It allows a simple addition of different organs in the simulation environment, especially when more experimental data on other organs involving the presented constitutive law will be available.

4.3 Implementation of the Non-Linear MSS Force in TopoSim

As we saw in Ch. 3, the force formulation applies to all of the springs of MSS by the *add_force* function in the `Spring` class. To modify the formulation of the forces it simply requires one to change the formula used in this function to a new one, which uses already available parameters, like the spring elongation dl, initial spring length l_0, the elongation velocity $dl/\ dt$, spring's stiffness k, spring damping d, etc. The three new parameters used by our non-linear force formulation (see Eq. (4.4)) have to be additionally defined as the *add_force* function arguments to easily change them depending on the type of the body. They will include the linear constitutive index n, hardening constitutive index p and visco-elastic strain limit γ_{lim}. From now on they are treated as the simulation input parameters alongside Young's modulus, Poisson's ratio, time step, body density and integration method. The implementation of the new formulation of force is very easy. Additionally, the simplicity of adjustments allows us to use this formulation for different types of tissues and reproduce various simulations.

4.4 Results of Simulations Using Non-linear Force Formulation

Let us start the presentation of our simulations' results from the linear and non-linear formulation of forces. We begin by verifying the linear approach, followed by the results of non-linear force formulation applied to simulate real tissues.

The simulation's objective is to measure the shear stress of the deformed material. It is done by constraining the bottom layer of the sample and applying a constant velocity (based on the strain rate $d\gamma/dt$) to the top layer. When the deformation occurs, we measure the shear stress response of the top layer to the applied deformation at every time-step.

The classical formulation of force in MSS (see Eq. (3.3), page 36) provides very little possibilities to control the stress-strain curve. The main source of the tissue's behavior is then the formulation of spring stiffness, which in our initial implementation incorporates the Young modulus and Poisson ratio as the physical parameters in the MSS. Using it we can try to simulate the linear (small, under the strain limit) deformations, but, as we show in this section, it is not sufficient and requires a use of a force formulation, which takes into account the particular behavior of selected tissue.

The mesh used for these simulations is in a shape of a cube consisting of $10 \times 10 \times 10$ hexahedral elements of a total size $5 \times 5 \times 5$ cm. The other parameters setup include density $\rho = 1000$ kg/m^3 and the gravity $g = 0$ m/s^2. Time step of the simulation was set to $dt = 0.001$ s. The topological aspect of the simulation is equivalent to the one described further on, in the Sec. 5.3. The parameters used for the linear simulation are limited to the Young modulus $E = 1000$ Pa and Poisson ratio $\nu = 0.499$ (indicating incompressibility of the simulated body and describing only elasticity of a material), which are in the ranges usually describing soft materials like internal organs. From this simulation we expect only an approximately correct result in comparison with the analytical solution.

The non-linear simulation works well with the parameters E, n and p chosen according to Tab. 4.1 from Ch. 4.1.2. The consistency K is used as a reference point to assign Young's modulus to the simulation input parameters. The only parameter which needs to be adjusted manually to fit the simulation to the analytical model is the strain limit γ_{lim}, which for our non-linear MSS we denoted γ_{lim_mss}. Strain limit is responsible for the part of the curve, which marks the start of the non-linear behavior. The reason for the adjustment of the strain limit and such an approximation of the Young modulus is simple

- our non-linear force formulation is only inspired by the constitutive law of Nicolle et al. and does not directly derive from it, so the use of the parameters is relative to our formulation. In our model it is the only parameter which value does not correspond to the strain limit of the strain hardening bi-power law derived by Nicolle et al. We adjust is manually to fit the resulting curve. The values used for this strain limit for our MSS non-linear simulations are therefore presented in Tab. 4.2.

Tissue	γ_{lim_mss}
Kidney	0.07
Liver	0.45
Spleen	400.00

TABLE 4.2: The values of γ_{lim_mss} used for the three types of internal organs in our MSS simulations.

4.4.1 Comparison of Linear and Non-Linear Formulations

As we have already seen, the two plots presented in Fig. 4.3 show, that the readability of the analytical formulation for the three strain rates $d\gamma/dt$ (being 0.0151 s^{-1}, 0.133 s^{-1}, 0.67 s^{-1}) is much better if we depict the results not as the stress-strain relationship (overlapping curves), but stress-time. Knowing the strain rates (change of strain per second), we can easily translate them to the time in seconds by just dividing the strain data γ by the strain rate $d\gamma/dt$. For that reason the presented results will be depicted in the form of stress-time relationship.

Fig. 4.4 and 4.5 show a simulation of a simple linear deformation of a kidney using MSS in comparison with the theoretical linear bi-power law (Eq. (4.3)). The linear deformation results show a correct alignment to the analytical results, however with a slope being different than the expected theoretical one, since there is no way to control the curve. This results in the relative errors for the three strain rates being very large.

To simulate real bodies with better precision we used the non-linear formulation of the force, presented in Ch. 4.2 in Eq. (4.4). Remembering Tab. 4.1 (given in Ch. 4.1.2) and Tab. 4.2 in which we noted all the necessary parameters to simulate three real organs (liver, kidney and spleen) we can proceed with presentation of the non-linear deformation results. We compare these results with analytical solution derived by Nicolle et al. [124, 123], (see Eq. (4.1)).

Fig. 4.6 presents the results of kidney simulation compared with the constitutive law. We can see, that the alignment with the theoretical solution gives us very satisfying result. We are able to control the shape of the curve by using the same parameters as the strain-hardening bi-power law is in addition to the MSS strain limit γ_{lim_mss}.

The same alignment takes place in case of liver and spleen simulation, which are depicted respectively in Fig. 4.7 and Fig. 4.8. The best alignment results are obtained each time for the $d\gamma/dt = 0.0151$ s^{-1}, however we can note, that the non-linear part of the curve is always fitted very well to the theoretical result. The linear parts of the curves are slightly translated in comparison with the analytical curves, but with a slope fitting exactly the expectations. The reason for the translation has its source in the new force formulation and the impact of the deformation speed on the curve, which is not big enough to fit perfectly the curve to the analytical solution.

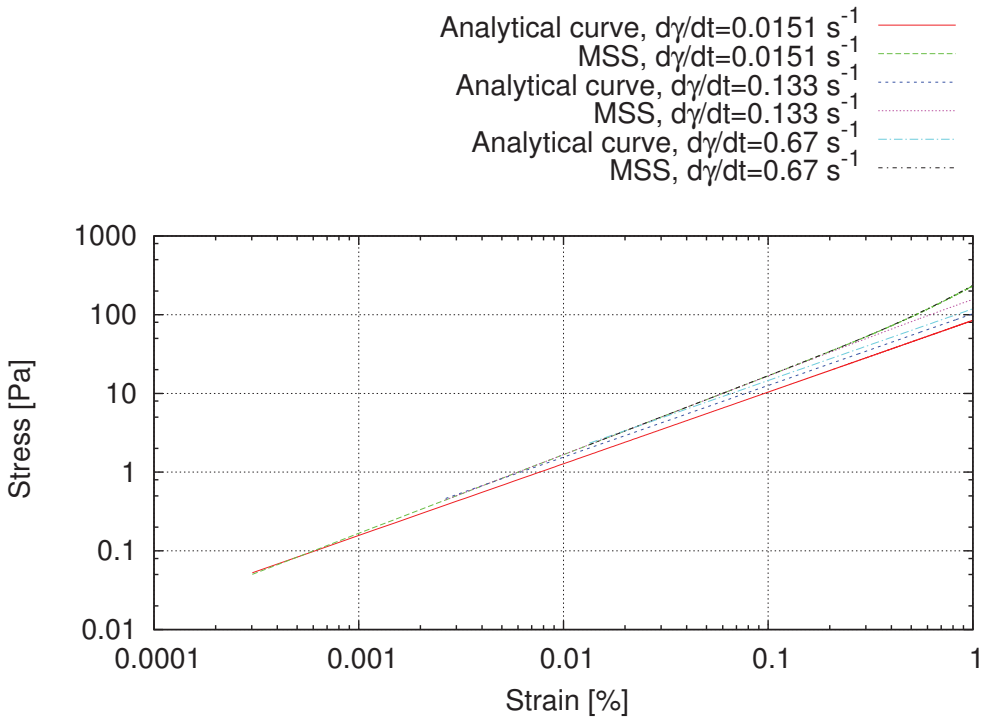


FIGURE 4.4: Stress-strain curve of kidney deformation obtained by the MSS with the linear formulation of forces, using three different strain rates $d\gamma/dt$, in comparison to the theoretical linear bi-power law (Eq. 4.3).

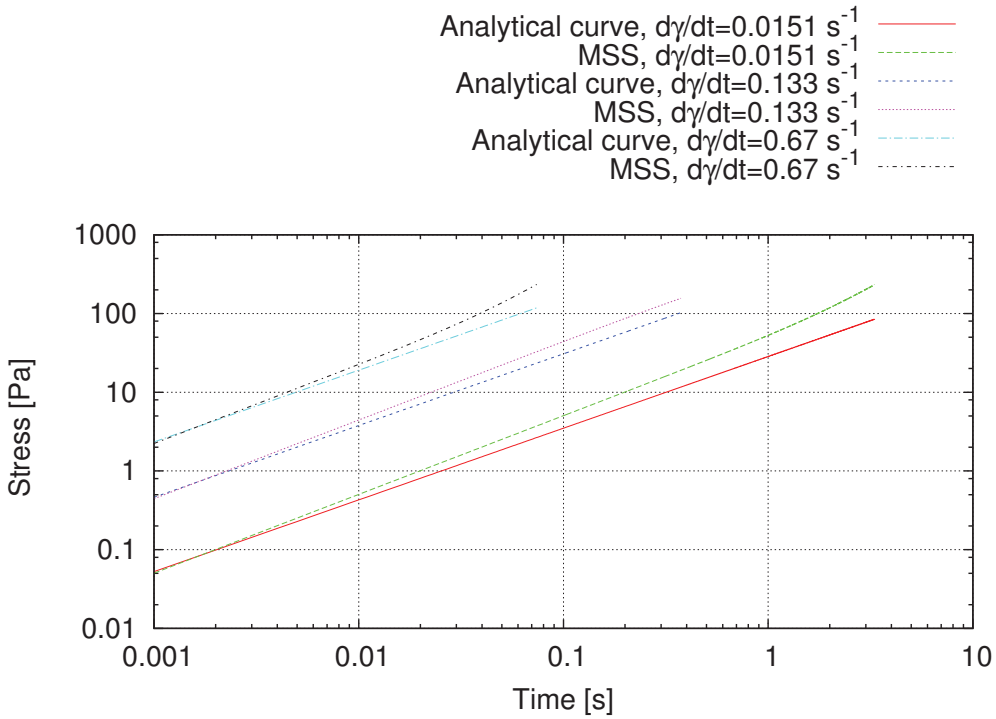


FIGURE 4.5: Stress-time curve of kidney deformation obtained by the MSS with the linear formulation of forces, using three different strain rates $d\gamma/dt$, in comparison to the theoretical linear bi-power law (Eq. 4.3).

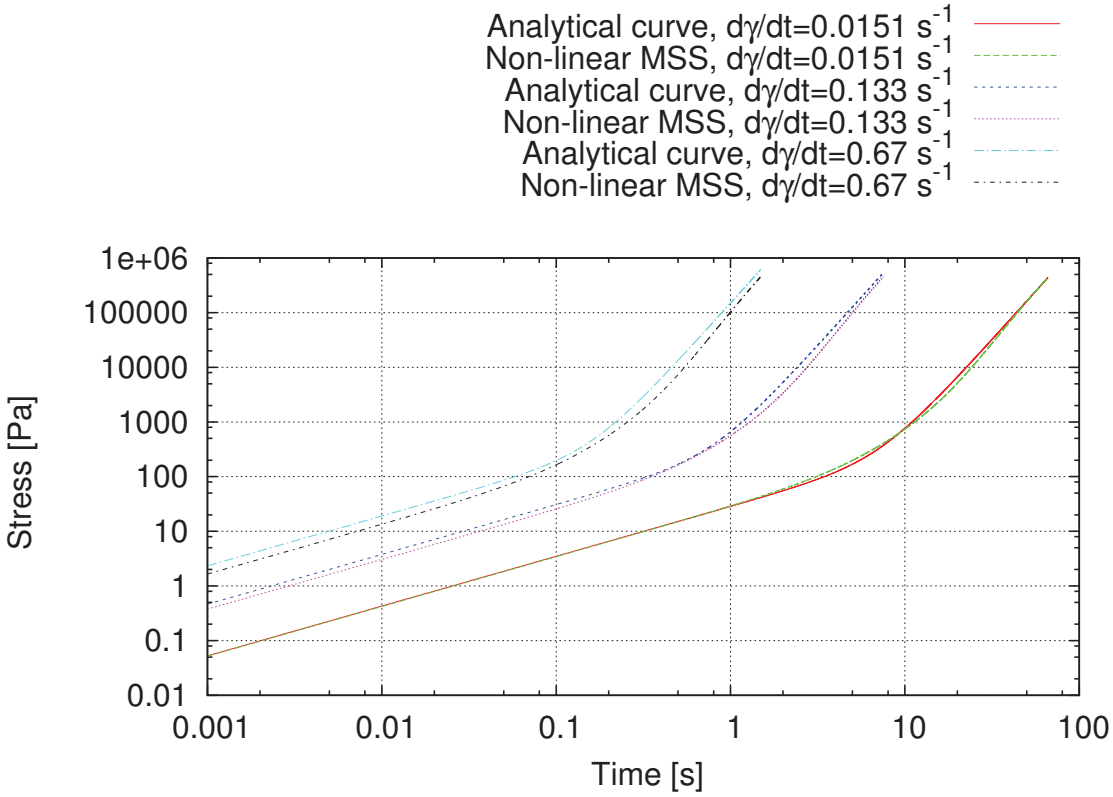


FIGURE 4.6: Non-linear MSS simulation of kidney at three different strain rates in comparison with analytical curve.

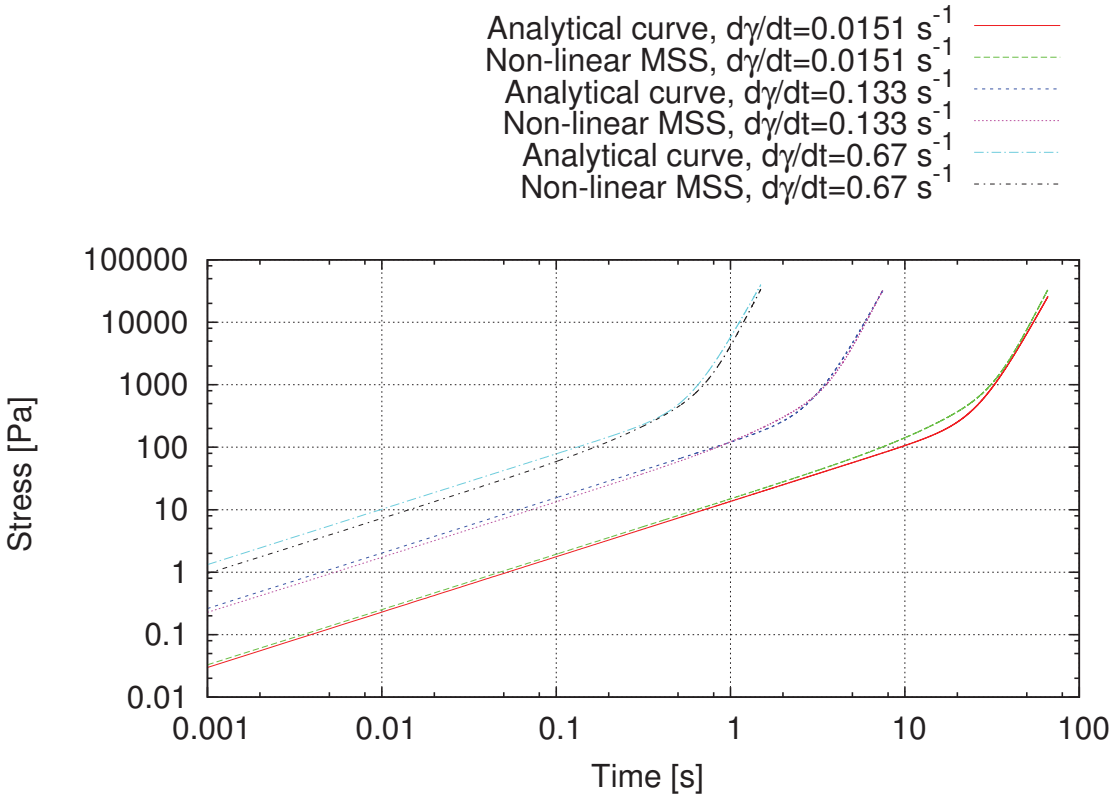


FIGURE 4.7: Non-linear MSS simulation of liver at three different strain rates in comparison with analytical curve.

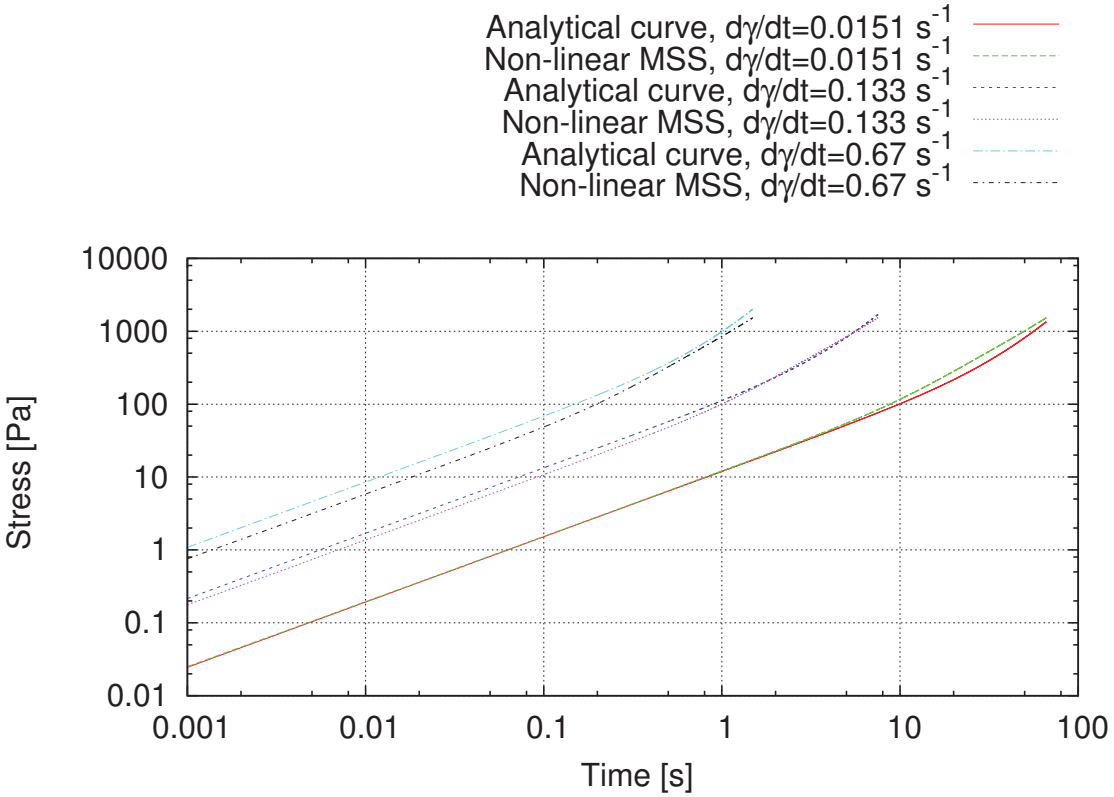


FIGURE 4.8: Non-linear MSS simulation of spleen at three different strain rates in comparison with analytical curve.

4.5 Discussion

In our experiments we successfully reproduced the non-linear visco-elastic behavior of liver, kidney and spleen tissues. We have shown that it is possible to control the deformation of a body and its force response by using a different than classical formulation of force and by adjusting the parameters of a given tissue. Thanks to those we can easily simulate its behavior almost exactly like it was done in the real experiments performed by Nicolle et al. [124, 123, 122]. The formulation is suited well for the visco-elastic tissues and it still leaves the room for modifications depending on the type of expected behavior *e.g.* incompressibility or non-linear behavior in different directions than just a simple shear. With a correct approach it is possible to add to the formulation term based on other deformation types, for example plastic deformation. By using the strain hardening bi-power law, having obtained the correct K, n and p parameters for different tissues and adjusting the γ_{lim_mss} parameter, we can obtain physically-based results of non-linear MSS simulation comparable to the constitutive law for different strain rates.

The improvements of the formulation include better adjustment to the theoretical bi-power law. We would like to be able to use all the parameters used by the analytical solution, so the improved formulation shall take into account the theoretical value of γ_{lim} instead of its manually adjusted counterpart. Additionally, the incompressibility of the material is not modeled correctly despite use of $\nu = 0.499$ with the presented initial MSS. Indeed, the presented experiments were based on a computation of the spring stiffness presented in Ch. 3, which is physically correct only for $nu = 0.25$. The value of the Poisson ratio we used for our non-linear experiments was $\nu = 0.499$ which did affect the simulation, however the occurred error was not corrected in regard with this parameter. Due to that error we have to improve our model to adjust the compressibility rate with the appropriate use of Poisson's ratio.

Chapter 5

Improvement of MSS for stable simulation with any Poisson's Ratio

As we have already seen, the existing simulation models are improved to compute the behavior of real tissues faster and with a higher precision. However, when working in the biomedical field - which requires robustness, high accuracy in terms of deformation and very often information about acting forces or haptic feedback - the MSS did not rise up to the challenge.

In Ch. 3 we saw that the MSS has several issues to provide a precise simulation. The first issue is visible especially in large deformations. It concerns the MSS instability related to wrinkles visible on the surface of a deformed body (Fig. 5.1). It is rarely recognized because of its occurrence: it strongly depends on the spring configuration and the magnitude of the deformation.

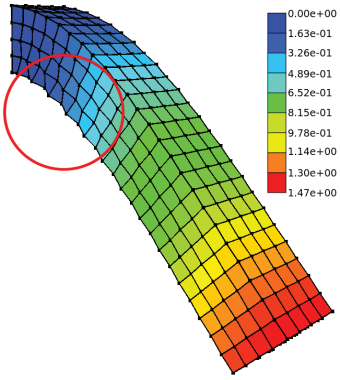


FIGURE 5.1: Wrinkle instability occurring at the base of a beam under gravity.

The second issue concerns the Cauchy limitation described by the elasticity theory, which is re-discovered by the scientists over and over again [12, 92, 91, 4]. It appears when we want to compute spring parameters and forces according to the mechanical properties of soft tissues. Indeed, the Poisson ratio must be set to a specific value to model an isotropic material. This value is $\nu_{2D} = 1/3$ in 2D and $\nu_{3D} = 1/4$ in 3D models. Additionally the existing solutions are strongly dependent on the type of the deformation (*i.e.* suitable only for specific loads like uniaxial tension or shear stress) and spring configuration.

Despite so many great improvements of MSS that are being made over the years, both limitations of MSS persist. We present here a new formulation of MSS which enables the use of the full range of Poisson's ratio, especially in case of isotropic bodies, and which allows to avoid wrinkle occurrence. It consists of two parts: formulation of the 3 spring

stiffness parameters according to the Young Modulus E of the material and the addition of volume correction forces based on the Poisson ratio ν of the material with $\nu \in [0; 0.5[$. At the end, we present assembly of our cubical MSS elements which allows to simulate a soft elastic material defined by its elastic parameters E, ν and its density $\rho = m/V_0$ with m the global mass of the material and V_0 its initial volume.

5.1 Definition of the Spring Stiffness

From Ch. 3.2 we know that the wrinkle and buckling instabilities occur in MSS if only the edge and inner-diagonal springs are used. In fact wrinkles occur systematically when there are no face-diagonal springs. However, adding face-diagonal springs introduces one additional parameter in the relationship between stiffnesses and Young modulus, and leads to an infinity of solutions for the values of stiffnesses. By choosing some solutions instead of other, we can push back the limit of appearance of wrinkles as will be demonstrated below.

Thus, we propose a cubical MSS with a configuration fulfilled by the face-diagonal springs. As shown in Fig. 5.2, we consider a full 3D cubical MSS element consisting of 12 springs on the edges (of stiffness constant k_e and length l_e), 4 inner diagonal springs placed inside the volume (of stiffness constant k_c and length l_c) and 12 face diagonal springs (of stiffness constant k_f and length l_f). Naturally, we have the following relations between the lengths of the springs: $l_e = l_f/\sqrt{2} = l_c/\sqrt{3}$.

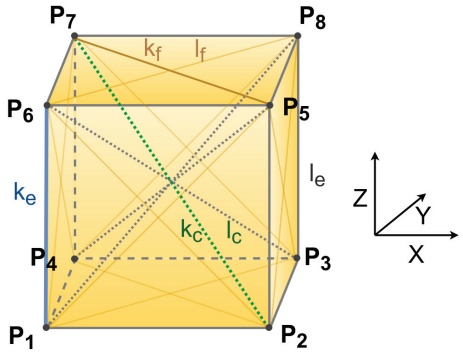


FIGURE 5.2: A 3D cubical MSS element composed of 12 edge springs, 12 face diagonal springs and 4 inner diagonal springs.

Using our 3D cubical MSS, a material is represented by a regular lattice with identical fundamental cubical elements wherein springs connect point-masses as presented beforehand. Moreover any mass is a center of symmetry of the lattice and, when no external forces act on it, the system takes a stable rest configuration. Note that by adding the face-diagonal springs in 3D we add stability to the system. Indeed the particles are not so free to move in the undesired directions any more and with the correct choice of stiffness parameters the wrinkles will not appear at all. Then we have to find the formulation of stiffness parameters (k_e, k_f, k_c) integrating the physical parameters of the object (Young's modulus E and Poisson's ratio ν).

To define the stiffness parameters of our hybrid cubical mass-spring system while ensuring correct mechanical behavior of the simulated material, we consider the global energy $\mathbf{W}_{system}$ of our system.

In the following, we first present the formulation of the stiffness constants k_e, k_c, k_f of the springs coming from the energy $\mathbf{W}_{springs}$ (defined by Eq. (3.1), page 37).

General Case

Let us first consider a MSS constructed using identical springs defined by a stiffness constant k and rest length l. We introduce $\mathbf{r}$ and $\mathbf{r}'$ of respective lengths l and l' as the extension vectors of the springs at rest and in a deformed state.

Let us apply a homogeneous deformation to the lattice of this MSS, whereby point $\mathbf{X}$ is displaced to $\mathbf{x} = \mathbf{X} + \mathbf{u}$. The displacement is described by $u_i = x_i - X_i = \partial u_i/\partial X_j \; X_j$, with $i, j = x, y, z$, where the displacement gradient $\partial u_i/\partial X_j = \nabla u_{ij}$ is small with respect to unity. By the assumption that any mass point is the center of symmetry for all springs connected to it, any spring is connected to identical springs, all with the same vector $\mathbf{r}$ aligned through the entire lattice, the spring vector too varies by a displacement $\delta = \nabla\mathbf{u}\cdot\mathbf{r}$ and its length varies by $\delta = l' - l = \sqrt{(\mathbf{r} + \nabla\mathbf{u}^T \cdot \mathbf{r})^2} - l$. At first order included in $\nabla\mathbf{u}$, we have:

$$\delta = \sqrt{(\mathbf{r} + (\nabla\mathbf{u})^T \cdot \mathbf{r})^2} = \frac{1}{2l}\nabla_i u_j r_i r_j \tag{5.1}$$

To the same precision, the elastic energy of the spring (Eq. (3.1), page 36) can be written:

$$W_{spring}(\delta) = \frac{1}{2}\nabla_i u_j \; \nabla_k u_l \; \frac{k}{l^2} \; r_i \; r_j \; r_k \; r_l. \tag{5.2}$$

Using the linearized strain tensor of Green-Lagrange, we have to keep in mind that the total elastic energy density of a fundamental cell defined in Eq. (3.6) (page 37), is obtained by summing over all springs α of the cell (being a unit cube), with:

$$C_{ijkl} = \frac{1}{V} \sum_{\alpha \in \text{cell}} \frac{k}{l^2} \; r_{\alpha\, i} \; r_{\alpha\, j} \; r_{\alpha\, k} \; r_{\alpha\, l}, \tag{5.3}$$

where $r_{\alpha\, i}$ is the i-th cartesian component of vector $\mathbf{r}_\alpha$. We perform a summation over i to l being implied, whereas summation over α identifying the springs is explicitly stated.

Our 3D Cubical MSS's Element

Now let us consider one cubical element of our MSS which is composed of 12 edge springs (k_e, l_e), 12 face diagonal springs (k_f, l_f) and 4 inner diagonal springs (k_f, l_f) as illustrated in Fig. 5.2 and which lays in the middle of an infinite lattice of the same cubical elements.

- Four of the edge springs are parallel to any of the axes directions, but each spring belongs to four adjacent cells, resulting into only one edge spring (k_e, l_e) to be counted in each direction. These are represented by the following vectors of identical length equal to l_e:

$$\mathbf{r_1} = (l_e, 0, 0)^T, \mathbf{r_2} = (0, l_e, 0)^T, \mathbf{r_3} = (0, 0, l_e)^T. \tag{5.4}$$

- The four inner diagonal springs are represented by the following vectors of identical length $l_c = \sqrt{3}\; l_e$:

$$\begin{aligned}\mathbf{r_4} &= (l_e, l_e, l_e)^T, \mathbf{r_5} = (l_e, l_e, -l_e)^T,\\ \mathbf{r_6} &= (l_e, -l_e, l_e)^T, \mathbf{r_7} = (-l_e, l_e, l_e)^T.\end{aligned} \tag{5.5}$$

- There are two identical vectors appearing on two opposite faces of the cell, but since each face belongs to two adjacent cells, one must count once each of the following vectors of identical length $l_f = \sqrt{2}\, l_e$:

$$\begin{aligned}\mathbf{r_8} &= (l_e, l_e, 0)^T, \mathbf{r_9} = (l_e, -l_e, 0)^T, \mathbf{r_{10}} = (0, l_e, l_e)^T,\\ \mathbf{r_{11}} &= (0, l_e, -l_e)^T, \mathbf{r_{12}} = (l_e, 0, l_e)^T, \mathbf{r_{13}} = (l_e, 0, -l_e)^T.\end{aligned} \tag{5.6}$$

Considering these assumptions, Eq. (5.3) for an element of our MSS becomes:

$$\begin{aligned}C_{ijkl} &= l_e^{-3}\left(\sum_{\alpha=1}^{3} l_e^{-2}\ k_e\, r_{\alpha i}\, r_{\alpha j}\, r_{\alpha k}\, r_{\alpha l} + \sum_{\alpha=4}^{7} l_c^{-2}\, k_c\, r_{\alpha i}\, r_{\alpha j}\, r_{\alpha k}\, r_{\alpha l}\right.\\ &\quad \left. + \sum_{\alpha=8}^{13} l_f^{-2}\, k_f\, r_{\alpha i}\, r_{\alpha j}\, r_{\alpha k}\, r_{\alpha l}\right)\\ &= l_e^{-5}\left(k_e \sum_{\alpha=1}^{3} r_{\alpha i}\, r_{\alpha j}\, r_{\alpha k}\, r_{\alpha l}\right.\\ &\quad \left. + \tfrac{1}{3}\, k_c \sum_{\alpha=4}^{7} r_{\alpha i}\, r_{\alpha j}\, r_{\alpha k}\, r_{\alpha l} + \tfrac{1}{2}\, k_f \sum_{\alpha=8}^{13} r_{\alpha i}\, r_{\alpha j}\, r_{\alpha k}\, r_{\alpha l}\right).\end{aligned} \tag{5.7}$$

From this equation, we directly obtain:

$$\begin{aligned}C_{xxxx} &= l_e^{-5}\left(k_e \sum_{\alpha=1}^{3} (r_{\alpha x})^4 + \tfrac{1}{3}\, k_c \sum_{\alpha=4}^{7} (r_{\alpha x})^4 + \tfrac{1}{2}\, k_f \sum_{\alpha=8}^{13} (r_{\alpha x})^4\right)\\ &= l_e^{-1}\left(k_e + \frac{4}{3}\, k_c + 2\, k_f\right),\end{aligned} \tag{5.8}$$

$$\begin{aligned}C_{xxyy} &= l_e^{-5}\left(k_e \sum_{\alpha=1}^{3} (r_{\alpha x} r_{\alpha y})^2 + \tfrac{1}{3}\, k_c \sum_{\alpha=4}^{7} (r_{\alpha x} r_{\alpha y})^2 + \tfrac{1}{2}\, k_f \sum_{\alpha=8}^{13} (r_{\alpha x} r_{\alpha y})^2\right)\\ &= l_e^{-1}\left(\frac{4}{3}\, k_c + k_f\right)\end{aligned} \tag{5.9}$$

Then, we have to satisfy Eq. (3.11) which give the relation between the elastic coefficients for an isotropic material modeled by a cubical MSS. Cauchy relation $C_{xxyy} = C_{xyxy}$ is automatically satisfied, as expected for a system of springs. Thus, let us remember that we have:

$$\begin{aligned}C_{xxxx} &= C_{xxyy} + 2\, C_{xyxy},\\ C_{xxyy} &= C_{xyxy} = \lambda = \mu\end{aligned} \tag{5.10}$$

leading to:

$$\begin{aligned} k_e &= \frac{8}{3}\, k_c + k_f \ , \\ \lambda = \mu &= \left(\frac{4}{3}\, k_c + k_f\right) l_e^{-1}. \end{aligned} \tag{5.11}$$

Consequently, to compute a formula for each stiffness coefficient k_e, k_f, k_c we still need one more dependency. For this purpose we decided that there exists a relation $k_f/k_c = A$ that will serve as a degree of freedom stabilizing the critical movements of the particles. After the computations, we get the following formulas for the stiffness constants, remembering that Cauchy equality forces us to use $\nu = 1/4$ (involving $\lambda = \mu = 2/5\ E$):

$$\begin{cases} \dfrac{k_f}{k_c} &= A \\ k_c &= \dfrac{3\ \mu\ l_e}{3A+4} = \dfrac{3\ E\ l_e}{10+7.5\ A} = \dfrac{3\ E\ l_c}{\sqrt{3}\ (10+7.5\ A)} \\ k_f &= \dfrac{3\ \mu\ l_e}{3+4/A} = \dfrac{3\ E\ l_e}{7.5+10/A} = \dfrac{3\ E\ l_f}{\sqrt{2}\ (7.5+10/A)} \\ k_e &= k_f + \dfrac{8}{3}\, k_c = \dfrac{3\ E\ l_e}{7.5+10/A} + \dfrac{8\ E\ l_e}{10+7.5\ A} \end{cases} \tag{5.12}$$

Assuming a case in which $A = 0$ the term $k_f = 0$, which means that we do not consider springs on the diagonals of the faces. Similarly, assuming $A \to \infty$ would mean that the terms k_c and $1/A$ will go to 0. In this case, we would have:

$$\begin{aligned} &\frac{k_f}{k_c} \to \infty \ ; \quad k_c \to 0 \ ; \\ &k_f = k_e = \frac{3\ E\ l_e}{7.5} = \frac{3\ E\ l_f}{7.5\ \sqrt{2}}. \end{aligned} \tag{5.13}$$

To sum up, considering one cubical element of the lattice, we obtain the formulations of the stiffness constants for the 3 kinds of springs involved in our MSS. These formulations are defined for any value of Young's modulus E and obeying the Cauchy restriction $\nu = 1/4$. It corresponds to the ratio k_f/k_c established according to the geometrical modeling of the material. In our experiments presented in Ch. 5.6 we have verified the impact of the value of A and we found out that a variation of $A \in [0.01; 100]$ does not impact the simulation in a *noticeable* way. The most visible impact that the parameter has on the simulation took place in the beam under gravity simulation case (see Fig. 5.3). Of course it is worth to note that extreme values of A: $A = 0$ or $A \to \infty$ result in using only two types of springs instead of all three. Nevertheless, by introducing A in our MSS we introduce a new degree of freedom which permits to better control the instability of our simulation method.

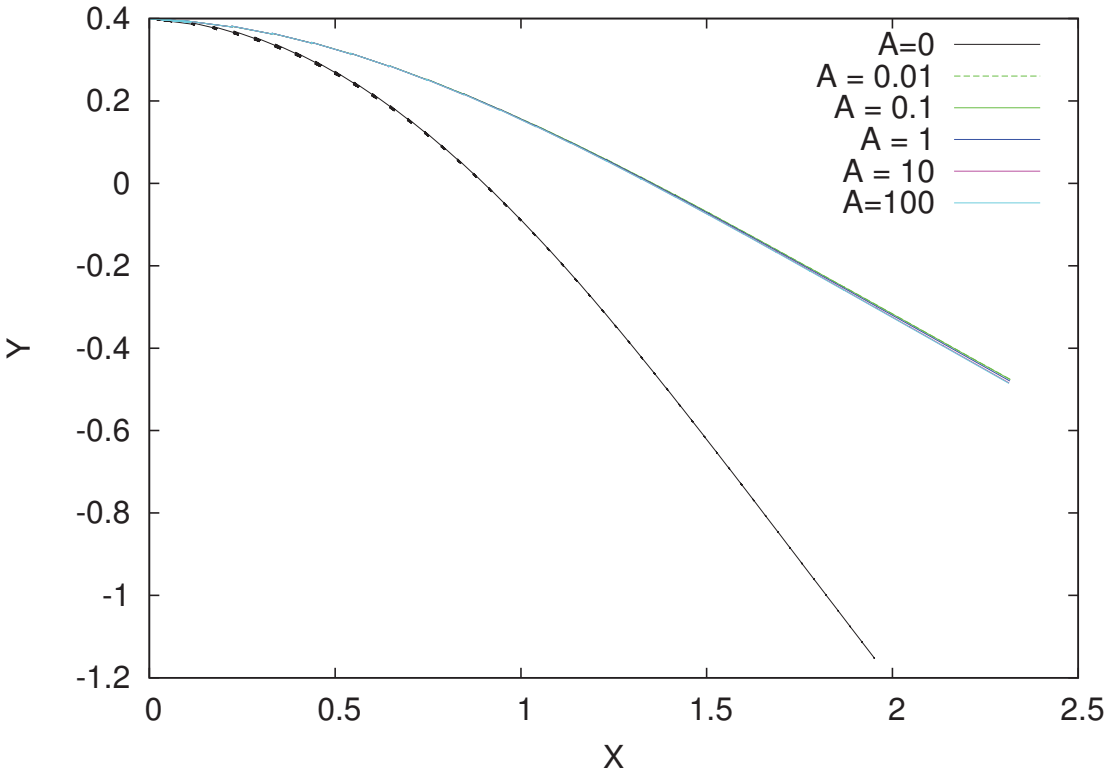


FIGURE 5.3: Comparison of MSS-CF beam under gravity experiments using different values of A parameter: $A = \{0.01, 0.01, 1, 10, 100\}$. $E = 2$ MPa, $\nu = 0.3$, $g = -10$.

5.2 Definition of the Volume Correction Forces

The second contribution of this chapter is finding a solution to escape the Cauchy limitation (imposing a Poisson ratio fixed to $1/4$) using a cubical MSS to simulate isotropic elastic material. As a solution we propose to add another type of forces - called correction forces - as an addition to the existing linear spring forces. At the end of each time step of the simulation loop the forces applied on each particle of our hybrid MSS correspond to the sum of the classical spring forces and the correction forces.

A similar idea was already proposed by [12], but our formulation ensures the use of any Poisson's ratio thanks to quadratic formulation of the correction forces. This means that we increase the degree of the formulation of forces of our system. Moreover, these additional forces will enable to control the volume variation of the material while considering its compressibility property (Poisson's ratio). These forces will ensure the correct volumetric behavior of the body without affecting the global shape of the deformation.

5.2.1 Energy of Volume Variation

To formulate these correction forces, we consider the energy due to the volume variation of the material during its deformation. To achieve this, let us consider a cubical element c of our system. We note its current volume V^c and its initial volume V_0^c. The coefficient κ of the material is linked to the volume variation and will be defined later on. The energy of our element c due to this variation is then defined by:

$$W_{vol}^c = \frac{1}{2}\,\kappa\,\frac{(V^c - V_0^c)^2}{V_0}, \tag{5.14}$$

with the corresponding energy density:

$$\widetilde{W}_{vol}^c = \frac{W_{vol}^c}{V_0^c} = \frac{1}{2}\,\kappa\,\left(\frac{V^c - V_0^c}{V_0^c}\right)^2 \tag{5.15}$$

and the following differential $\mathrm{d}W_{vol}^c = \cdots \mathrm{d}V^c$ of the energy:

$$\delta W_{vol}^c = \kappa\,\frac{V^c - V_0^c}{V_0}\,\delta V^c. \tag{5.16}$$

From the energy density we can deduce the corresponding correction force that we have to apply on each particle of the element in order to control its volume variation. Thus, considering a particle P_i residing in a position $\mathbf{P}_i$, its correction force is defined by:

$$\mathbf{F}_{P_i} = -\frac{\partial W_{vol}^c}{\partial \mathbf{P}_i} = -\kappa\,\frac{V^c - V_0^c}{V_0}\,\frac{\partial V^c}{\partial \mathbf{P}_i}. \tag{5.17}$$

Consequently, to compute this force, we need: (i) to estimate the volume V^c of the element and its gradient $\partial V^c/\partial \mathbf{P}_i$ according to each particle P_i, all during the simulation loop; (ii) to express the coefficient κ with E and ν.

5.2.2 Approximation of the Current Volume

As illustrated in Fig. 5.2, we consider a cubical element composed of 8 particles noted P_i for $i \in \{1,\ ...,\ 8\}$. After some deformations this element is no more a cube, but it is slightly distorted. We can compute three vectors $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ to describe a mean value of the cube's edges in Cartesian directions. These vectors are computed by the following intuitive formulas:

$$\begin{cases} \mathbf{X} = \frac{1}{4}(\overrightarrow{\mathbf{P}_1\mathbf{P}_2} + \overrightarrow{\mathbf{P}_4\mathbf{P}_3} + \overrightarrow{\mathbf{P}_6\mathbf{P}_5} + \overrightarrow{\mathbf{P}_7\mathbf{P}_8}) \\ \mathbf{Y} = \frac{1}{4}(\overrightarrow{\mathbf{P}_2\mathbf{P}_3} + \overrightarrow{\mathbf{P}_1\mathbf{P}_4} + \overrightarrow{\mathbf{P}_5\mathbf{P}_8} + \overrightarrow{\mathbf{P}_6\mathbf{P}_7}) \\ \mathbf{Z} = \frac{1}{4}(\overrightarrow{\mathbf{P}_1\mathbf{P}_6} + \overrightarrow{\mathbf{P}_3\mathbf{P}_8} + \overrightarrow{\mathbf{P}_2\mathbf{P}_5} + \overrightarrow{\mathbf{P}_4\mathbf{P}_7}). \end{cases} \tag{5.18}$$

Then, considering a linear deformation of the volume, instead of a complex and time consuming computation of splitting each hexahedron into tetrahedra, we can get an approximation of the current volume V^c using these 3 vectors with:

$$V^c = \mathbf{X} \cdot (\mathbf{Y} \times \mathbf{Z}) = \mathbf{Y} \cdot (\mathbf{Z} \times \mathbf{X}) = \mathbf{Z} \cdot (\mathbf{X} \times \mathbf{Y}). \tag{5.19}$$

5.2.3 Formulation of the Correction Forces

Now we can express the formulation of the correction forces for each particle P_i of a cubical element with $i \in [1, 8]$. Considering Eq. (5.17) and computing the volume gradient $\partial V^c / \partial \mathbf{P}_i$ according to Eq. (5.19), we obtain the following formulas:

$$\begin{aligned}
\mathbf{F}_{P_1} &= -\frac{\kappa\,(V^c - V_0^c)}{4\,V_0}\,(-(\mathbf{Y}\times\mathbf{Z}) - (\mathbf{Z}\times\mathbf{X}) - (\mathbf{X}\times\mathbf{Y}))\,, \\
\mathbf{F}_{P_2} &= -\frac{\kappa\,(V^c - V_0^c)}{4\,V_0}\,((\mathbf{Y}\times\mathbf{Z}) - (\mathbf{Z}\times\mathbf{X}) - (\mathbf{X}\times\mathbf{Y}))\,, \\
\mathbf{F}_{P_3} &= -\frac{\kappa\,(V^c - V_0^c)}{4\,V_0}\,((\mathbf{Y}\times\mathbf{Z}) + (\mathbf{Z}\times\mathbf{X}) - (\mathbf{X}\times\mathbf{Y}))\,, \\
\mathbf{F}_{P_4} &= -\frac{\kappa\,(V^c - V_0^c)}{4\,V_0}\,(-(\mathbf{Y}\times\mathbf{Z}) + (\mathbf{Z}\times\mathbf{X}) - (\mathbf{X}\times\mathbf{Y}))\,, \\
\mathbf{F}_{P_5} &= -\frac{\kappa\,(V^c - V_0^c)}{4\,V_0}\,((\mathbf{Y}\times\mathbf{Z}) - (\mathbf{Z}\times\mathbf{X}) + (\mathbf{X}\times\mathbf{Y}))\,, \\
\mathbf{F}_{P_6} &= -\frac{\kappa\,(V^c - V_0^c)}{4\,V_0}\,(-(\mathbf{Y}\times\mathbf{Z}) - (\mathbf{Z}\times\mathbf{X}) + (\mathbf{X}\times\mathbf{Y}))\,, \\
\mathbf{F}_{P_7} &= -\frac{\kappa\,(V^c - V_0^c)}{4\,V_0}\,(-(\mathbf{Y}\times\mathbf{Z}) + (\mathbf{Z}\times\mathbf{X}) + (\mathbf{X}\times\mathbf{Y}))\,, \\
\mathbf{F}_{P_8} &= -\frac{\kappa\,(V^c - V_0^c)}{4\,V_0}\,((\mathbf{Y}\times\mathbf{Z}) + (\mathbf{Z}\times\mathbf{X}) + (\mathbf{X}\times\mathbf{Y}))\,,
\end{aligned} \tag{5.20}$$

with the logical relations:

$$\mathbf{F}_{P_1} = -\mathbf{F}_{P_8},\ \ \mathbf{F}_{P_2} = -\mathbf{F}_{P_7},\ \ \mathbf{F}_{P_3} = -\mathbf{F}_{P_6},\ \ \mathbf{F}_{P_4} = -\mathbf{F}_{P_5}. \tag{5.21}$$

5.2.4 Definition of κ.

To compute all the necessary equations, we have to express the coefficient κ according to the actual material properties: Poisson ratio ν and the Young modulus E. Keep in mind that V_0 and V correspond respectively to the initial and current volume of the entire model. Moreover, if we consider $\boldsymbol{\epsilon}$ as the linearized strain tensor of Green-Lagrange, we know that $Tr(\boldsymbol{\epsilon})$ corresponds to the volume variation of the material during the simulation with $Tr(\boldsymbol{\epsilon}) = (V - V_0)/V_0$.

Then the energy density of our hybrid MSS with correction forces is defined by:

$$\widetilde{W}_{system} = \frac{W_{system}}{V_0} = \frac{W_{springs}}{V_0} + \frac{W_{vol}}{V_0} = \widetilde{W}_{springs} + \widetilde{W}_{vol}. \tag{5.22}$$

First let us focus on the energy due to the springs by only considering the 3D MSS without correction forces. Simulating an isotropic elastic material we know from Eq. (3.9) and the Cauchy equality that its elastic energy density is:

$$\widetilde{W}_{springs} = \frac{\mu}{2}(Tr(\boldsymbol{\epsilon}))^2 + \mu\, Tr(\boldsymbol{\epsilon}^2). \tag{5.23}$$

Then, if we focus on the energy density at a volume variation, we can note from Eq. (5.15) and the expression of $Tr(\boldsymbol{\epsilon})$ that we have:

$$\widetilde{W}_{vol} = \frac{1}{2}\,\kappa\,(Tr(\boldsymbol{\epsilon}))^2. \tag{5.24}$$

At the end, considering our complete hybrid MSS with the correction forces and recalling that $(V - V_0)/V_0 = Tr(\boldsymbol{\epsilon})$, its energy density is defined by:

$$\begin{aligned}\widetilde{W}_{system} &= \widetilde{W}_{springs} + \widetilde{W}_{vol} \\ &= \frac{1}{2}\,(\mu+\kappa)\,(Tr(\boldsymbol{\epsilon}))^2 + \mu\, Tr(\boldsymbol{\epsilon}^2)\end{aligned} \tag{5.25}$$

The corresponding stress tensor is then defined by:

$$\sigma_{ij} = \frac{\partial \widetilde{W}_{system}}{\partial \epsilon_{ij}} = (\mu+\kappa)\,\delta_{ij}\,\epsilon_{kk} + 2\,\mu\,\epsilon_{ij}. \tag{5.26}$$

Now, let us consider an uniaxial strain experiment performed on our isotropic, elastic material. This experiment is characterized by $\sigma_{11} = \sigma_{22} = \sigma_{33} = 0$ and $\sigma_{ij} = 0$ for $(i,j) \neq (1,1)$. Thus, we obtain the following expressions for the stress tensor components:

$$\left\{\begin{array}{lcl}\sigma_{11} &=& (3\mu+\kappa)\,\epsilon_{11} + (\mu+\kappa)(\epsilon_{22}+\epsilon_{33}) \\ \sigma_{22} &=& (3\mu+\kappa)\,\epsilon_{22} + (\mu+\kappa)(\epsilon_{11}+\epsilon_{33}) = 0 \\ \sigma_{33} &=& (3\mu+\kappa)\,\epsilon_{33} + (\mu+\kappa)(\epsilon_{11}+\epsilon_{22}) = 0 \\ \sigma_{ij} &=& \sigma_{ji} = 2\mu\,\epsilon_{ij},\ i \neq j.\end{array}\right. \tag{5.27}$$

From these we can deduce that $\epsilon_{22} = \epsilon_{33}$. Consequently, the previous expressions can be reformulated:

$$\left\{\begin{array}{lcl}\sigma_{11} &=& (3\mu+\kappa)\,\epsilon_{11} + 2\,(\mu+\kappa)\,\epsilon_{22} \\ \sigma_{22} &=& \sigma_{33} = (\mu+\kappa)\,\epsilon_{11} + 2\,(2\mu+\kappa)\,\epsilon_{22} = 0 \\ \sigma_{ij} &=& \sigma_{ji} = 2\mu\,\epsilon_{ij},\ i \neq j.\end{array}\right. \tag{5.28}$$

Using Eq. (5.27) will result in:

$$\sigma_{11} = \frac{\mu\,(5\,\mu + 3\,\kappa)}{2\,\mu+\kappa}\,\epsilon_{11}, \tag{5.29}$$

and from Eq. (5.28) we obtain:

$$\epsilon_{22} = -\frac{\mu+\kappa}{2\,(2\mu+\kappa)}\,\epsilon_{11} = \epsilon_{33}, \tag{5.30}$$

Then, knowing that the Poisson ratio and the Young modulus are defined for this type of body by $\nu = -\epsilon_{22}/\epsilon_{11}$ and $E = \sigma_{11}/\epsilon_{11}$, we have:

$$\nu = \frac{\mu + \kappa}{2\ (2\mu + \kappa)}, \tag{5.31}$$

$$E = \frac{\mu\ (5\ \mu + 3\ \kappa)}{2\ \mu + \kappa}. \tag{5.32}$$

By inverting these relationships, we obtain:

$$\mu = \frac{\kappa(2\ \nu - 1)}{1 - 4\ \nu}, \tag{5.33}$$

$$\kappa = \frac{\mu(5\mu - 2E)}{(E - 3\mu)} \tag{5.34}$$

finally leading to the following relations:

$$\mu = \frac{E}{2\ (1 + \nu)}, \tag{5.35}$$

$$\kappa = \frac{1}{2}\frac{E\ (4\nu - 1)}{(1 + \nu)(1 - 2\nu)} = K - \frac{5}{3}\ \mu. \tag{5.36}$$

If we observe the formulation of the coefficient κ (Eq. (5.36)) depicted in Fig. 5.4 as a function of the Poisson ratio $\kappa/E\ (\nu)$, we note the following:

- for $\nu = 1/4$ the κ coefficient is equal to 0 leading to $\widetilde{W}_{vol} = 0$, since we do not need correction force for this kind of material (as the 3D cubical MSS naturally modeled it);
- for $\nu \to 1/2$, the κ coefficient moves toward the infinity, as infinite forces have to be applied to preserve the volume of the incompressible material;
- for $\nu = 0$, $\kappa = -\frac{1}{2}\ E$. Thus, in the compression experiment we can observe no dilatation in the transverse direction and the change of the volume is equal to the displacement of the top face of the fully compressible material;
- although all known materials are characterized by $\nu \geq 0$, the stability limits of ν are $-1 \leq \nu \leq 1/2$.

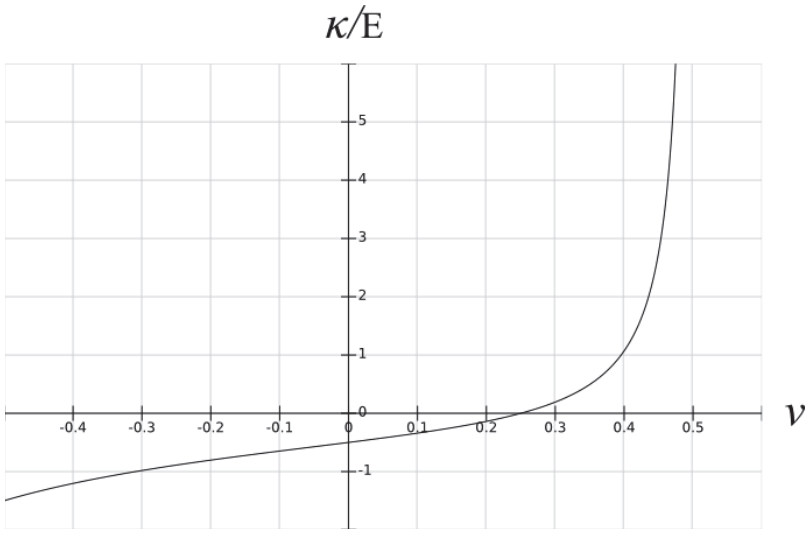


FIGURE 5.4: The κ coefficient depicted as a function of the ratio κ/E in terms of the Poisson ratio ν.

5.3 Assembly of Cubical MSS Elements

To simulate the mechanical behavior of a complex object, we represent it by an assembly of our 3D hybrid MSS elements. To create the MSS, we proceed as following:

- From the application of Kirchhoff's law we can deduce the values of spring stiffnesses at the sample boundaries. The edge springs which are common for n neighboring cubical elements are represented by a stiffness $k_e\ n/4$, which is adequately smaller than stiffness of springs which lay in the middle of the lattice (*i.e.* with four neighboring cubical elements).
- We consider the same division for the face diagonal springs on the boundaries of the mesh, which are defined by $k_f/2$ if they do not share a face with any other element.
- During the simulation, the forces applied on each particle of the MSS are computed as the sum of the springs forces and the correction forces computed considering each element of the MSS.

At the end our hybrid MSS is straight-forward in the usage, it has adjustable parameters and succeeds to simulate incompressible isotropic materials. Additionally, the full range of Poisson's ratio values (meaning $[0; 0.5[$) can be used while simulating real isotropic bodies thanks to the formulation, which escapes Cauchy's limitations. There are no additional structures introduced into MSS other than its classical springs and the user is finally able to model the behavior of isotropic bodies with various Young's modulus and Poisson's ratio values.

5.4 Implementation of Our Hybrid MSS in TopoSim

Unlike the non-linear force implementation, the use of additional springs and moreover the correction forces require looking deeper into the structure of the framework.

5.4.1 *Face-Diagonal* Springs

Let us start by adding the face-diagonal spring to the existing MSS. From Fig. 3.18 (page 63) we know that in 2D models the face-diagonal springs are attributes of the 2-cells (objects of `Face_LCC_MSS` class) and in 3D the inner-diagonal springs are the attributes of 3-cells (objects of `Volume_LCC_MSS` class). Since our face-diagonal springs will be used only in 3D, we need to add them to the MSS structure in the correct corresponding classes. This will be done in three simple steps:

- Add 12 face-diagonal springs to each volume;
- Add initialization of those springs to assign correct stiffness values;
- Include the force computation in the *add_force* function for the new springs for the global accumulation loop.

Let us now verify these steps starting from the bottom layer of `LCC_MSS` from Fig. 5.6:

- The starting point is to add the *face-diagonal* springs as the attributes alongside the inner-diagonal springs. To do that we created separate structures and methods to assign, modify and get information about those particular springs in the `Volume_LCC_MSS` class. It is very important to know the spring extremities (particles they are connecting) and to be consistent when it comes to the order of numbering the springs - it will be crucial information to accumulate the forces on the particles further on.
- Then, to initialize the correct stiffness we needed to add proper code with the new stiffness formulation to the *initialize_springs* function in the `Object_LCC_MSS` class, which includes the face-diagonal springs of 3D elements.
- The last step is to modify the *add_force* function to include the new springs in the global force accumulation loop. As the *accumulate* function is called globally on the whole MSS from the `Object_LCC_for_simu` class, we just need to modify the contents of the local *add_force* function of the `Volume_LCC_MSS` and compute the forces of the face-diagonal springs the same way as we already do for the inner-diagonal springs of each volume.

5.4.2 Force Corrections

The next method to be implemented in our framework is to apply the correction forces. Let us take a look at the procedure of applying the force corrections properly in the following steps:

- Add volume attribute to the 3-cells information; add its initialization;
- Add global correction forces loop;
  - For all volumes compute the current volume;
  - For all particles of the volume accumulate the correction forces according to Eq. (5.17).

Information about the Volume

First of all we need to add the current and initial volume of a 3D element. Therefore we naturally need to add a *volume* attribute to the `Volume_LCC_MSS` class. Additionally, we will add the volume computation function to the same class and use it to compute the initial volume V_0 in the *initialize_volumes* function of the `Object_LCC_for_simu` class. Later on this function will serve to compute the current volume of a 3-cell.

Application of the Correction Forces

The loop applying the correction forces, which we called *correct_volumes*, needs to be added as the last step of the global *accumulate* loop. The reason for this is simple: the classical simulation loop will first accumulate the forces on the particles and then update their positions; if we wanted to apply the correction forces right after the first iteration we would have to call the *correct_volumes* as the third step of the simulation and then update the positions again. To avoid the unnecessary computations of the new positions we apply this step right after the *accumulate* loop (see Fig. 5.5). Let us present the implementation of the correction forces in the form of modified simulation loop in Algorithm 2.

Algorithm 2: TopoSim simulation loop algorithm

```
forall Objects do
    Accumulate
        forall Particles j do
            accumulate external forces: F_j ← F_ext;
        forall Springs i do
            compute forces F_i;
            accumulate forces on the extremity particles:
            F_p1 ← +F_i;
            F_p2 ← −F_i ;
        forall Volumes do
            correct_volumes()
    Simulation
        forall Particles j do
            compute the new positions x_j using an integration scheme given F_j
```

At the first iteration of the simulation loop the positions of the particles are not updated yet, so the correction forces will be equal to 0 for all 3-cells, however at the second iteration we will already know the new positions of all the particles and so will be able to compute the adequate force corrections accordingly. Note that the application of the correction forces is conditional: as we know from Eq. (5.17), the corrections will be equal to 0 if $\kappa = 0$. Therefore, as we remember from Eq. (5.36), $\kappa = 0$ when $\nu = 1/4$. Thus, we do not need to compute the corrections if the Poisson ratio is set to $1/4$.

Implementation of the Correction Forces

The next step is to implement the calculation of the force corrections in our program. Since the corrections are computed based on the change of the volume of a 3-cell, it is only natural to calculate them at the level of `Volume_LCC_MSS` class. The new *correct_volumes*

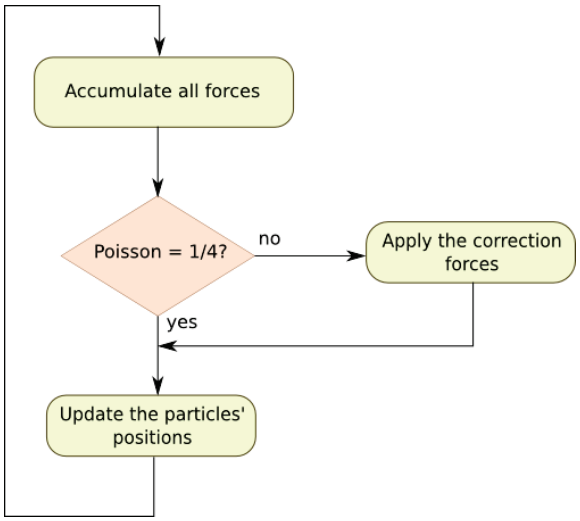


FIGURE 5.5: UML diagram of the algorithm to apply force corrections.

function will compute the current volume of the 3-cell and use it (together with the previously established κ and $V0$) to calculate and accumulate the correction forces on all 8 particles of the 3-cell.

All the changes and newly added attributes and structures are depicted in Fig. 5.6. We marked the new elements in green and the affected functions which needed modifications in blue. The evident drawback of the presented implementation approach of the correction forces is to iterate over the number of 3-cells (let us denote it N) and then again over all the particles of a hexahedron, which means that we need to serially accumulate forces $8 \times N$ times at every iteration. In this case, the simple parallelization of this calculation using the TBB method is not an obvious one, since we need to manage the access to the particles' information. If at one point the system wants to add forces to a particle p_i from two neighbouring 3-cells at the same time - the tasks running in parallel may run into an access conflict. For this reason we decided to parallelize this approach in the future work and currently use it in the serial version.

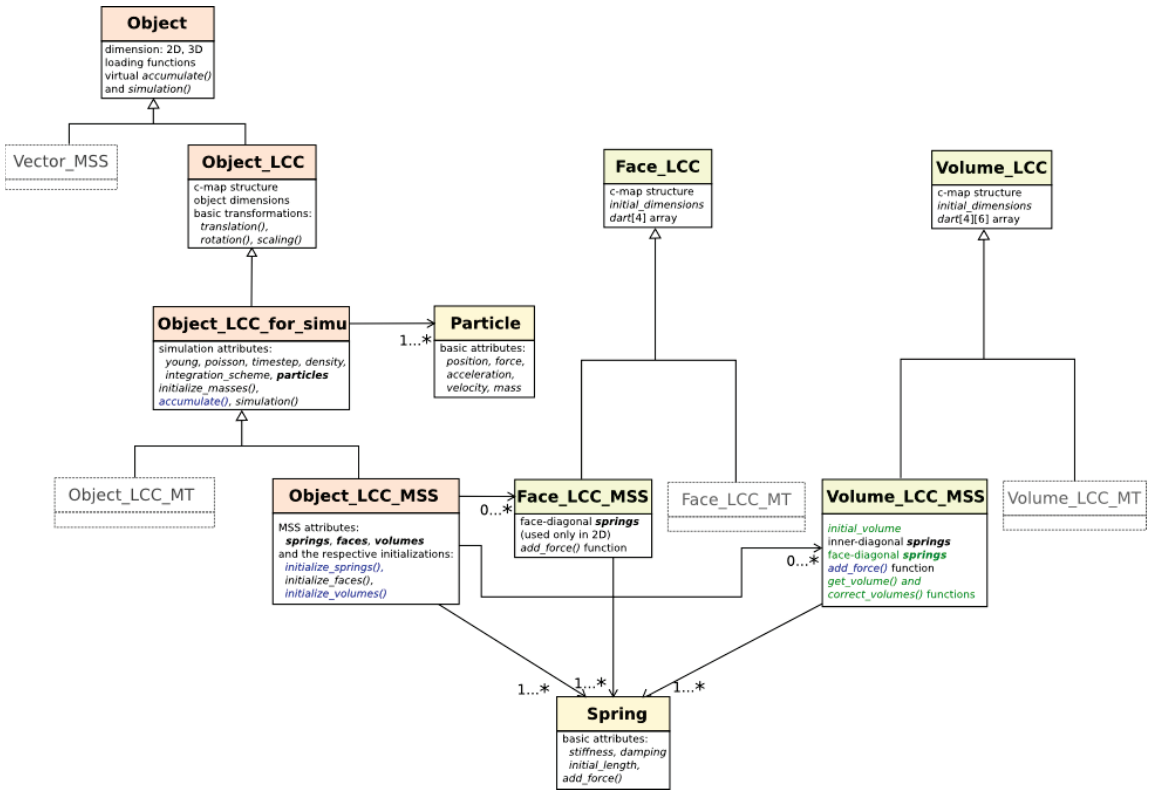


FIGURE 5.6: UML class diagram of the TopoSim basic simulation components after adding the face-diagonal springs and the force corrections. Newly added elements are marked in green and the modified functions in blue.

5.5 Expected Mechanical Behavior

As already mentioned, the Young modulus and Poisson ratio are parameters used to describe the elastic behavior of a body. Fig. 5.7 (top right) shows a simple 2D uniaxial tensile test, where on the top part of a square MSS there is an acting force **F**. The resulting deformation is elongating the square in the direction of the acting force by ζ. In the same time the face-diagonal springs of the MSS element cause the thinning of the shape by total amount of γ. The Poisson ratio and Young modulus are in such a case defined by:

$$\nu = \frac{\gamma l}{\zeta}, \quad E = \frac{\mathbf{F}}{\zeta}. \tag{5.37}$$

Let us look closer at the Poisson ratio and the behavior it determines. Looking at the bottom of Fig. 5.7 we can study two cases: absolute compressibility, which takes place when $\nu = 0$ and incompressibility, which occurs when $\nu = 1/2$. In the latter case the expected behavior of the body is to preserve the area of the deformed square, meaning that the area of the rectangle A will equal exactly the area of the undeformed square A_0: $A = (l_e + \gamma)(l_e - \zeta) = l_e^2 = A_0$. In the former case though, the compressibility of the shape has to be equal the compression rate of the vertical springs: $A = A_0(l_e - \zeta)$.

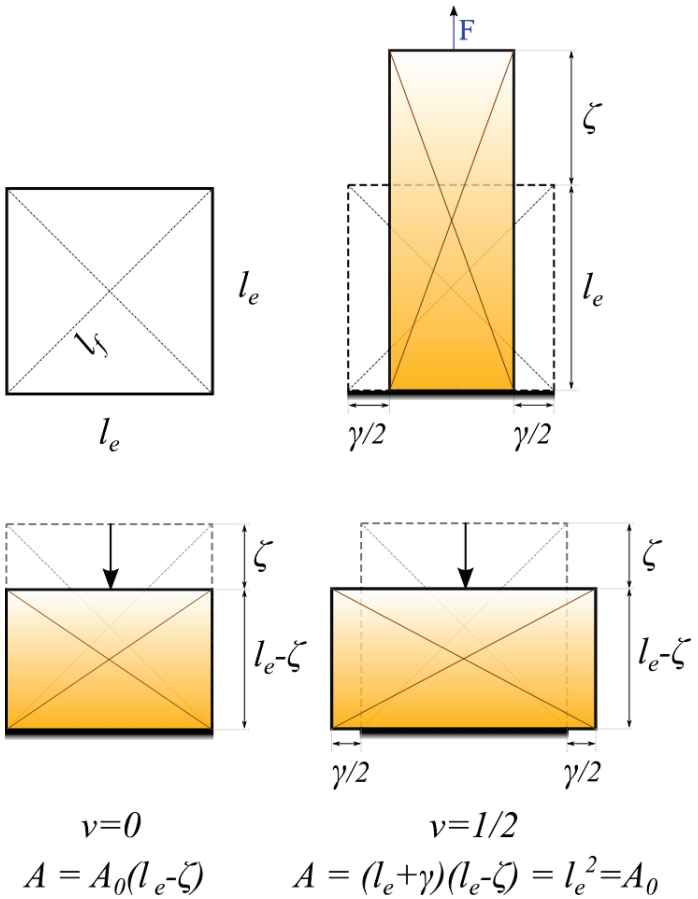


FIGURE 5.7: Correct elastic behavior of a square MSS element (top left) under tension (top right) and compression (bottom) loads.

We verified the mechanical behavior of a simple rectangular element at uniaxial compression and tension loads. Knowing the expected mechanical behavior of such a body

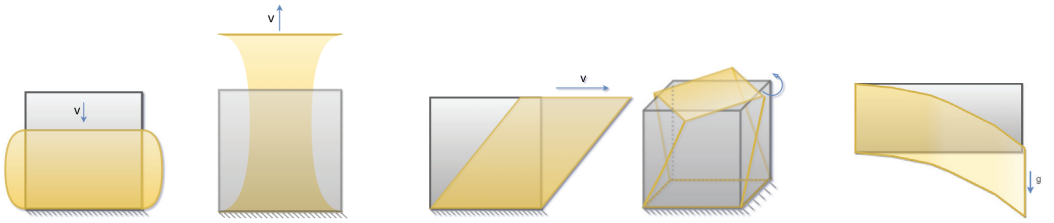
we can validate the experimental results performed using our MSS with volume correction forces.

5.6 Our Hybrid MSS Experimental Results

In this section we present the results of our newly proposed stiffness formulation and correction forces combined with use of all 3 types of MSS springs. To fully show the capabilities of our solution we compare experiments performed on our MSS with results obtained using other simulation approaches. As a complement to our validation we talk about the performance of our program in comparison to the previous approach and study the speed-up of the proposed method.

5.6.1 Experimental Setup

To validate the accuracy of implemented methods one can choose different types of comparison. First of all there is comparison with an analytical solution. That involves constitutive laws connected with mechanical properties or theoretical results which are directly associated with those. The comparison can also be done with another reliable method, like Finite Element Method. Especially good validation method when it comes to soft tissues is a comparison of the *response* of a tissue to different types of loads. Here we can compare the stress-strain or stress-time relationship when a load is applied and (but not necessarily) when the body comes back to its previous state (called relaxation phase) or evolution of torque in time for particular strain rates. In our validation we have compared our results with these obtained by other methods.



(a) Compression (b) Tension (c) Shearing (d) Torsion (e) Beam flexion

FIGURE 5.8: Five types of body deformation for a model validation.

Five Basic Deformations

These comparisons have been performed in five basic deformation types, as illustrated in Fig. 5.8:

Uniaxial compression. The first test case is compression (see Fig. 5.8a). It can be performed and studied in the following way: the top layer particles' velocities were imposed simulating applied pressure on the tissue while the bottom particles were constrained. This way one can control the speed of the deformation and study the force response of the body itself as well as its volume.

Uniaxial tension. The second test is tension (see Fig. 5.8b) which is similar to the compression experiment. For this test we constrained one of the walls and implied the movement (by imposing velocities) on the opposite wall's particles of the sample in the

perpendicular direction. During the deformation we studied the force response and volume change.

Shearing. The third deformation, depicted in Fig. 5.8c is shearing. Usually such an experiment is performed on a body which is stretched up to certain stresses or/and up to failure. We perform this experiment by constraining the bottom layer of the tissue and implying a change of the positions of the parallel layer at a constant speed - strain rate. As the comparable output of such an experiment one can use the stress-strain relationship.

Torsion. The fourth experiment with which it is possible to validate a model is torsion (see Fig. 5.8d). To perform this experiment, we rotated sample's top layer at a constant speed while the parallel bottom layer of the sample was constrained. The studied parameter in such experiment was torque.

Beam flexion. The last experiment is beam bending. For this test, a long beam was attached with its smallest face to an imaginary wall and then it was exposed to the gravity (see Fig. 5.8e).

Having such an experimental setup we are sure to cover vast majority of the possible scenarios of body deformations, which are usually a combination of the presented basic loads. Additionally, with available theoretical values and expected body behavior we can successfully verify our solution.

In this context we compared our results with the following methods:

- The Mass-Spring System (denoted MSS-Baudet) with the stiffness formulation proposed by Baudet [12] and defined in Ch. 3, using only two types of springs: edge and inner-diagonal;
- Our Mass-Spring System with the stiffness formulation defined in Eq. (5.1), using the correction forces, but with the parameter A set to 0 in order to simulate no face-diagonal springs (denoted MSS-nFD);
- Our Mass-Spring System with all springs and correction forces which is the full formulation presented in Ch. 5.1 (denoted MSS-CF);
- The Finite Element Method using LS-Dyna software (denoted FEM). These experiments are performed on LS-Dyna version 971dR4 (LSCT, Livermore, CA, USA) using the linear elastic material law of Hooke, fully integrated selective reduced solid elements and implicit numerical integration scheme.

Now let us present the details of each experiment from its technical point of view. The presented tests were performed using the following principles, assumptions and meshes:

- In all the instances (except of the beam bending experiment which is examined in the equilibrium state under gravity), we fixed the time-step to 0.001 s and the total number of iteration steps to $10{,}000$ (*i.e.* 10 s), without applying gravity force. Deformations were limited to 5% and 20% to verify the behavior of the simulated tissue at both small and big loads.
- For the correction forces expressed in Eq. (5.17), we used the compression modulus computed by Eq. (5.36), where $\nu \neq 1/2$. Note that in case of $\nu = 1/4$ these correction forces are equal to 0, *i.e.* the MSS naturally models this case.

- The tensile and the beam bending experiments were performed on a beam of a size $0.4 \times 0.4 \times 2.4$ cm^3. Two different mesh densities were used for both experiments: $4 \times 4 \times 24$ and $8 \times 8 \times 48$ elements. Additionally, for the beam flexion we use the gravity constant $g = -10$ m/s^2.
- All the other experiments were performed on a sample of $10 \times 10 \times 10$ cubes of a unit size 0.5 cm, which makes a sample of total volume $V_0 = 125$ cm^3.
- The mass density of all the MSS samples was set to $\rho = 1000$ kg/m^3 and the A parameter value to 1.
- The corresponding FEM experiments were performed on samples of size matching exactly the ones used for MSS tests.

Note that each of the five basic experiments was performed with same set of parameters and studied for different responses to show the usability of the system. Additionally the models we compared allow to see the impact of use of the correction forces and consequences of not using the face-diagonal springs with the comparison between MSS-nFD and MSS-CF.

5.6.2 Results of the Experiments Using Different Models

Compression

Let us start the comparison of the experiments from the compression test. The detailed parameters description and results after 10 seconds of deformation process are presented in Table 5.1. The column e shows the relative change of the volume computed using the following formula: $e = |V_0 - V^c|/V_0$ to validate body's volume behavior during simulations.

As we know from the compressibility rate, using $\nu = 0$ should produce results of a body which is wholly compressible. That means that the relative volume change e must equal the deformation. The FEM model represents the perfect values for both deformations of 5% and 20% while the MSS-CF model keeps up with a difference of volume change differing only by 0.61% and 0.12% relatively. We can also see that the volume change of MSS-Baudet model does not correspond to the expected values, while MSS-nFD improves these results, but with a bigger volume difference with comparison to FEM.

When modeling a material with $\nu = 1/4$ the expected volume change is not an obvious one, but we assume the FEM result to be the closest to the expected one. The value shall be close to half of the deformation rate and here again we can observe the MSS-CF being very close to the FEM result.

The last deformation case is $\nu = 0.499$, so nearly incompressible material, which should not change its volume at all. In this case we can observe that MSS-Baudet does not model this case properly while all the other simulation techniques manage to constrain the volume with a minimal relative volume change.

In Fig. 5.9, showing the compression experiments, we can see a great difference in the behavior of the samples. Note that the color scale corresponds to the displacement from the original position. The wrinkle instability is very noticeable in experiments on MSS-Baudet and MSS-nFD, which are clearly 'wrinkled'. In the same time, the MSS-nFD at $\nu = 0.499$, and MSS-CF and FEM at $\nu = 0.25$ and $\nu = 0.499$ samples show slightly 'swollen' but regular figure with divergent volume distribution. The conclusion we can draw from these observations is clear: without the use of face-diagonal springs we are sure to obtain results with wrinkle instability. The use of correction forces allows a better volumetric

behavior of the experiments. Incompressible part of this experiment ($\nu = 0.499$) shows slightly different image though. Unlike all the other experimental FEM cases, the FEM compression at $\nu = 0.499$ is also slightly 'wrinkled', which was definitely not an expected outcome. We assume this behavior to be a result of using a linear elastic law in the FEM simulation. Here (*i.e.* $\nu = 0.499$ experiments) the impact of the correction forces and use of all three types of springs is visible the best. Indeed we can observe a slight wrinkling close to the corners of the MSS-CF case. Nevertheless, the use of face-diagonal springs and volume correction forces gives the most regular figure result. Additionally we can verify the numerical results from Tab. 5.1 with the compressible ($\nu = 0$) MSS-CF experiment and see that the walls of the cube are perfectly straight, exactly as expected.

Model	**E**	ν	**deformation**	**e**	**instabilities**
MSS-Baudet	10,000	0	5%	2.6%	yes
MSS-nFD				5.37%	yes
MSS-CF				5.61%	no
FEM				5.00%	no
MSS-Baudet	10,000	0.25	5%	3.17%	yes
MSS-nFD				3.25%	yes
MSS-CF				3.17%	no
FEM				3.10%	no
MSS-Baudet	10,000	0.499	5%	3.49%	yes
MSS-nFD				0.00%	yes
MSS-CF				0.01%	no
FEM				0.02%	no
MSS	10,000	0	20%	12.99%	yes
MSS-nFD				23.62%	yes
MSS-CF				19.88%	no
FEM				20.00%	no
MSS-Baudet	10,000	0.25	20%	16.68%	yes
MSS-nFD				15.99%	yes
MSS-CF				13.52%	no
FEM				12.70%	no
MSS-Baudet	10,000	0.499	20%	18.73%	yes
MSS-nFD				0.01%	yes
MSS-CF				0.00%	no
FEM				0.10%	yes

TABLE 5.1: Testing parameters and results of the compression simulations.

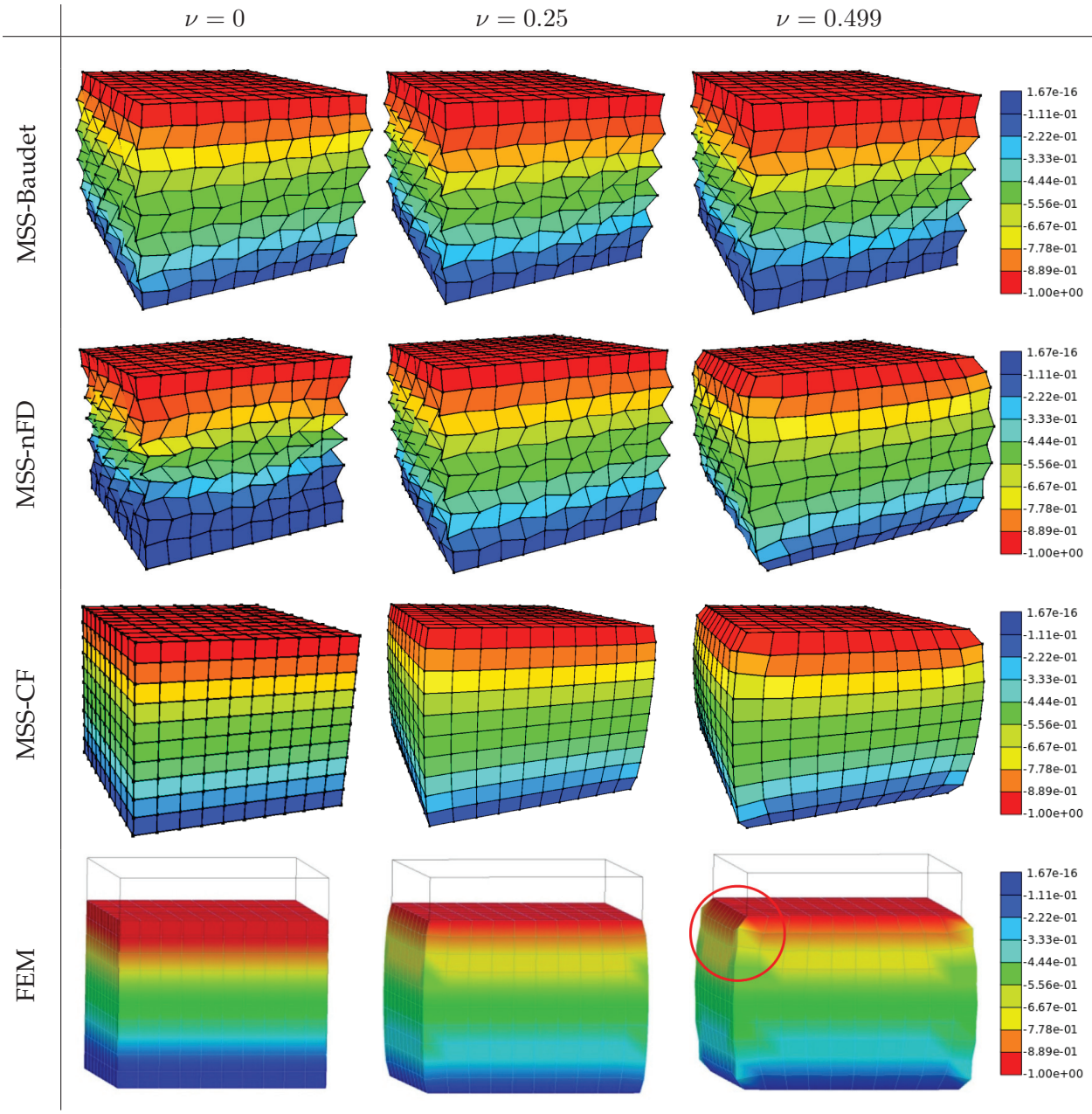


FIGURE 5.9: Compression experiment comparison between MSS-Baudet, MSS-nFD, MSS-CF and FEM at a big (20 %) deformation with different Poisson's ratios.

Tension

Table 5.2 presents the results of experiments performed with the four simulation models. Again, we can observe how the correction forces in MSS-CF influence the relative volume change making the results very close to the theoretical (in case of $\nu = 0$ and $\nu = 0.499$) and FEM (in case of $\nu = 0.25$) results. It is also visible that MSS-Baudet does not correspond to the expected values and additional instabilities make the outcome even worse.

Fig. 5.10 showing tensile test results is the opposite of the compression experiment. Here we can see that the volume of compressible MSS-CF ($\nu = 0$) has a very regular shape, with the horizontal walls staying parallel to the X axis, without any curve, similarly to the FEM solution with the same parameters and the MSS-nFD experiment. In turn, the MSS-Baudet at $\nu = 0$ case of the same deformation has a deep curve to it. We owe the correct shape of the other models to the correction forces, which 'guard' the correct volume changes within the samples.

Model	E	ν	deformation	e	instabilities
MSS-Baudet	10,000	0	5%	0.63%	yes
MSS-nFD				7.46%	no
MSS-CF				4.40%	no
FEM				5.00%	no
MSS-Baudet	10,000	0.25	5%	1.14%	yes
MSS-nFD				0.85%	no
MSS-CF				2.54%	no
FEM				2.61%	no
MSS-Baudet	10,000	0.499	5%	1.54%	yes
MSS-nFD				0.00%	no
MSS-CF				0.00%	no
FEM				0.00%	no
MSS-Baudet	10,000	0	20%	0.25%	yes
MSS-nFD				24.92%	no
MSS-CF				20.00%	no
FEM				19.99%	no
MSS-Baudet	10,000	0.25	20%	0.73%	yes
MSS-nFD				3.15%	yes
MSS-CF				8.90%	no
FEM				10.11%	no
MSS-Baudet	10,000	0.499	20%	1.13%	yes
MSS-nFD				0.00%	yes
MSS-CF				0.00%	no
FEM				0.00%	no

TABLE 5.2: Testing parameters and results of the tension simulations.

We can also observe that the distribution of the volumes is much improved between MSS-Baudet and MSS-CF for different ν values and MSS-nFD with $\nu = 0.25$ and $\nu = 0.499$. Here we can ensure this kind of shape distibution to the use of all types of springs, knowing that they are not used in MSS-Baudet and MSS-nFD models. Additionally, all MSS-nFD, MSS-CF and FEM with $\nu = 0.25$ and $\nu = 0.499$ show a nice flex curve, which is slightly too deep for the MSS-nFD simulation at $\nu = 0.25$. According to Tab. 5.2, even

the correction forces cannot change the huge impact, which is made by the lack of face-diagonal springs, making the variations between MSS-nFD and MSS-CF volumes noticeable.

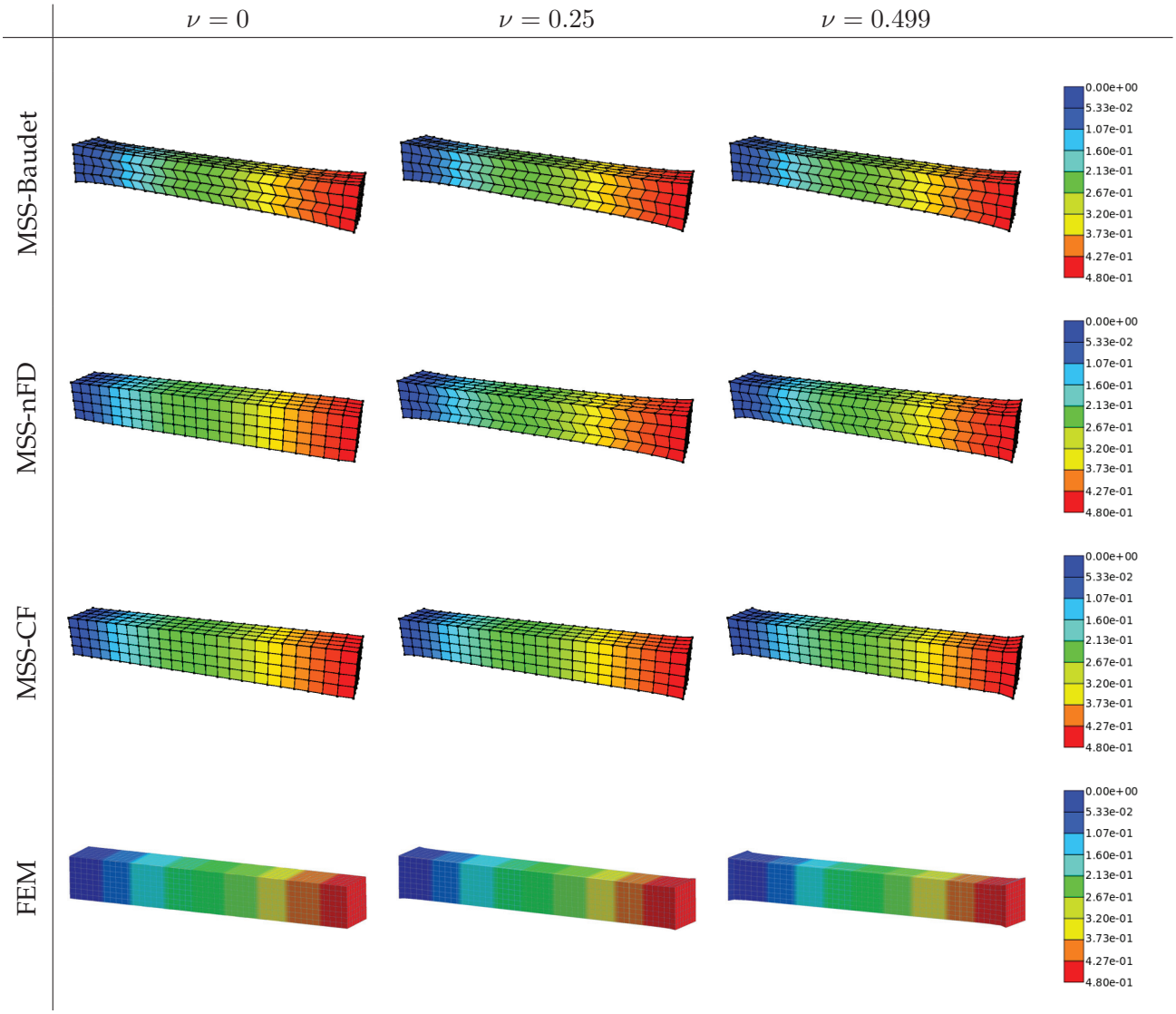


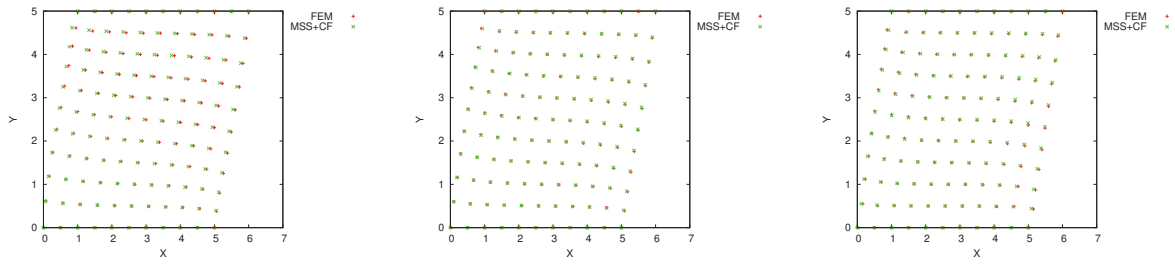
FIGURE 5.10: Tension experiment comparison between MSS-Baudet, MSS-nFD, MSS-CF and FEM with big (20 %) deformation.

In both experiments presented in Tables 5.1 and 5.2, the difference between the models' error is noticeable. The highest relative error's value of the volume conserving MSS-CF model ($\nu = 0.499$) reaches 0.01%; the biggest deviation of that simulation method from the FEM appeared in the 20% deformation in tensile test with $\nu = 0.25$ and it differs by 1.21%, which is a very acceptable error to say the least. We can easily see that the change of the MSS-CF volume in comparison to FEM has a very small deviation in general. However, it is essential to compare these numerical results with the actual shape deformations to see the real role of the correction forces and use of all types of springs.

Shearing

In case of the shearing experiment we can study the shape of the deformed body on Fig. 5.11 and 5.12. The differences between MSS-Baudet, MSS-nFD, MSS-CF and FEM simulations was not noticeable, therefore we present a comparison only between the latter two simulation techniques. Fig. 5.11 represents the cross-sectional view of the middle layer (meaning $Z = 5$) in the $X - Y$ plane. A slight wave on the layers of the sample, which is in fact a natural behavior of an elastic body, is an obligatory element of this type of deformation. Here, the correction forces ensure smoothness of the sample at every layer, and the difference between the MSS-CF and FEM is minor, since it is the role of the inner diagonal springs to deform the volumes correctly at this type of deformation.

We can also study the average difference of the MSS-CF and FEM points positions at this layer, which for the Poisson ratio values $\nu = 0$, $\nu = 0.25$ and $\nu = 0.499$ is respectively: 0.042 cm, 0.013 cm and 0.019 cm.



(a) $\nu = 0$ (b) $\nu = 0.25$ (c) $\nu = 0.499$

FIGURE 5.11: Alignment of cross-sectional view on points in a shearing experiment with big (20 %) deformation at different Poisson's ratios, $E = 10{,}000$; comparison between MSS-CF and FEM.

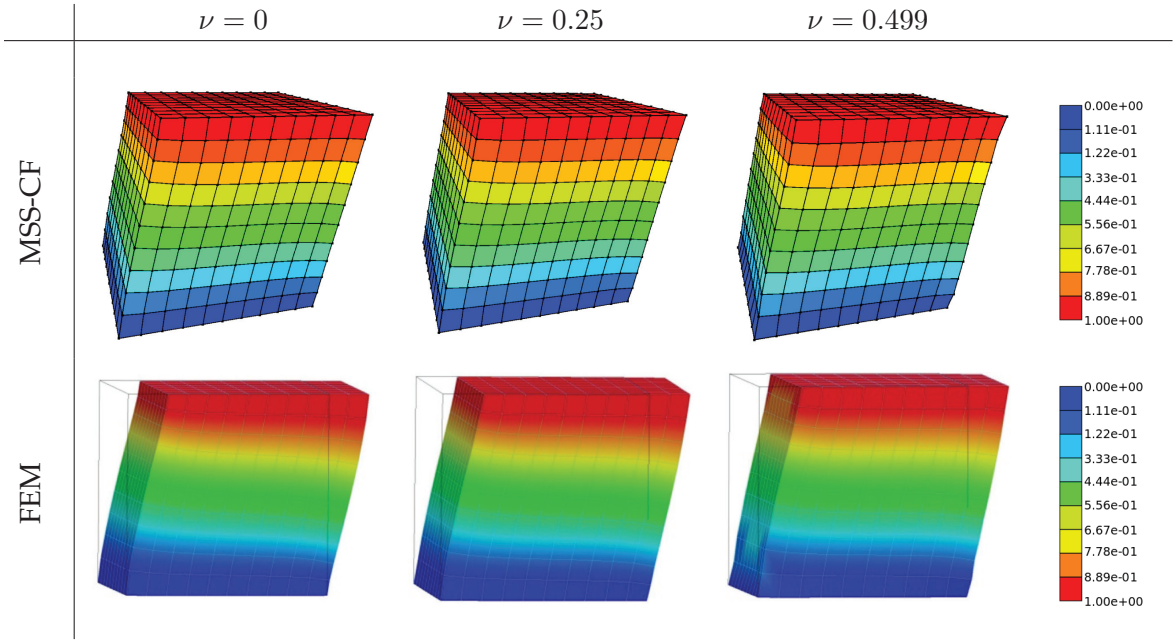


FIGURE 5.12: Shearing experiment with big (20 %) deformation at different Poisson's ratios, $E = 10{,}000$; comparison between MSS-CF and FEM.

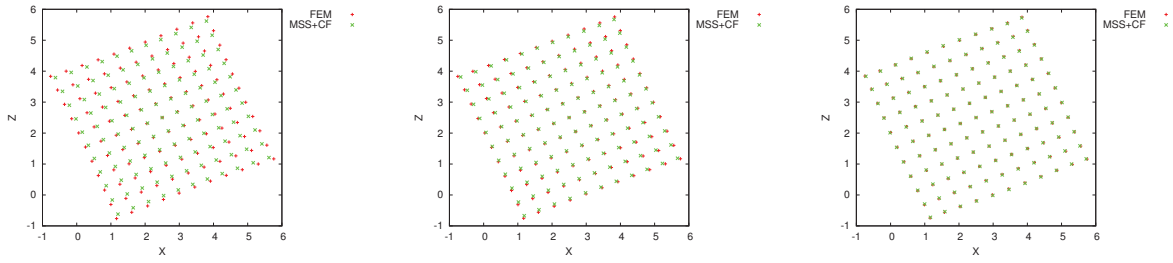
Additionally, Table 5.3 presents a study of the order of magnitude of the top plate response to deformation for shearing experiment with Poisson's ratio $\nu = 1/4$. The presented values represent force ([N]) and we can see that for MSS-nFD and MSS-CF they do not differ much from the FEM, unlike the MSS simulation method, which in turn is out of range of the other simulation methods. Clearly the use of correction forces allows to obtain realistic results, which agree with the Finite Element Simulation.

Deformation	MSS-Baudet	MSS-nFD	MSS-CF	FEM
5%	10,985	3,390	3,396	3,317
20%	46,460	14,662	14,600	13,234

TABLE 5.3: Response of the top plate in shearing deformation with $E = 10{,}000$, $\nu = 0.25$. The measured response is force [N].

Torsion

Adequately, Fig. 5.13 and 5.14 show the results of torsion experiment, where we present only MSS-CF in comparison with FEM due to the lack of visible differences with other models. In Fig. 5.13 we can compare the cross-sectional view of the middle layer ($Y = 5$) on $X - Z$ plane of a deformed body. The experiments are visibly smooth and we can see how close the MSS-CF is to the FEM solution. Here again we can verify the average difference of the MSS-CF and FEM point's positions to additionally confirm the MSS-CF model's accuracy. For the Poisson ratio values $\nu = 0$, $\nu = 0.25$ and $\nu = 0.499$ the average difference between points reaches respectively: 0.064 cm, 0.030 cm and 0.008 cm.



(a) $\nu = 0$ (b) $\nu = 0.25$ (c) $\nu = 0.499$

FIGURE 5.13: Alignment of cross-sectional view on points in a torsion experiment with big (20 %) deformation at different Poisson's ratios, $E = 10{,}000$; comparison between MSS-CF and FEM.

Similarly as in case of the shearing experiment, Table 5.4 presents a study of the order of magnitude of the top plate response to torsional deformation, also with Poisson's ratio $\nu = 1/4$. We can see that the order of magnitude of the MSS-Baudet and MSS-CF experiments correspond with the FEM solution, which we owe to the correction forces and physically based stiffness formulation.

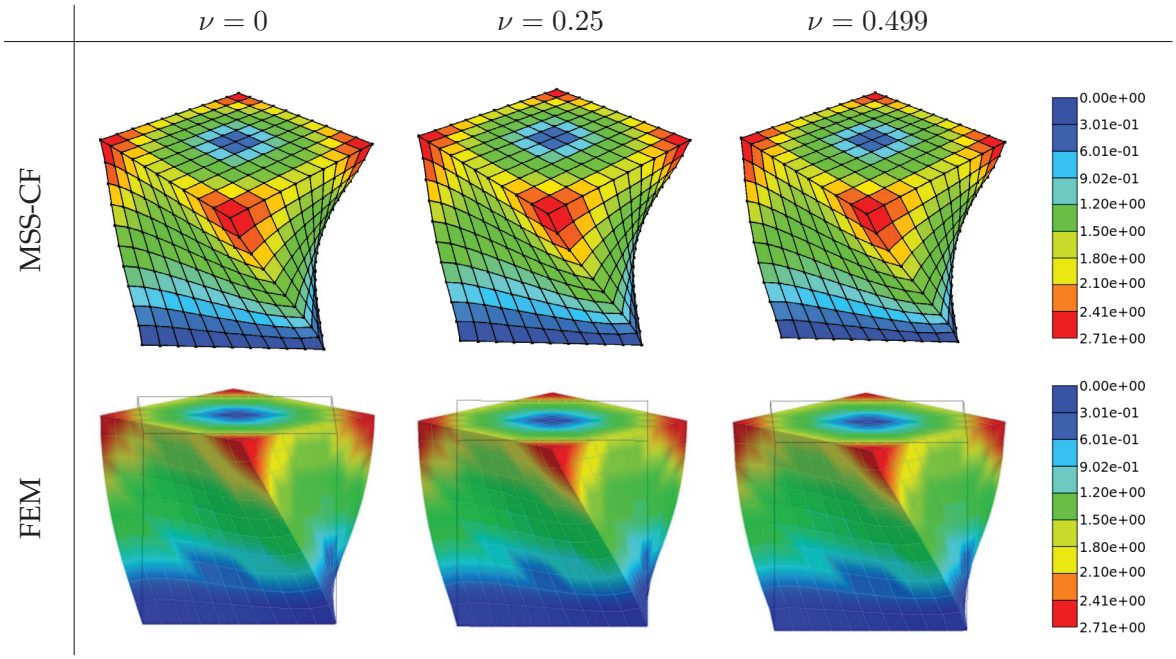


FIGURE 5.14: Torsion experiment with big (20 %) deformation at different Poisson's ratios, $E = 10{,}000$; comparison between MSS-CF and FEM.

Deformation	MSS-Baudet	MSS-nFD	MSS-CF	FEM
5%	3,463	1,277	1,306	1,279
20%	160,892	60,005	60,565	57,981

TABLE 5.4: Response of the top plate in torsion deformation with $E = 10{,}000$, $\nu = 0.25$. The measured response is torque [N·m].

Beam bending

Fig. 5.16 and 5.17 present the results of the beam under gravity for $E = 2$ MPa and $E = 10$ MPa, MSS-Baudet and MSS-nFD experiments in comparison with the analytical curves computed using Euler-Bernoulli equation which is defined by [105]:

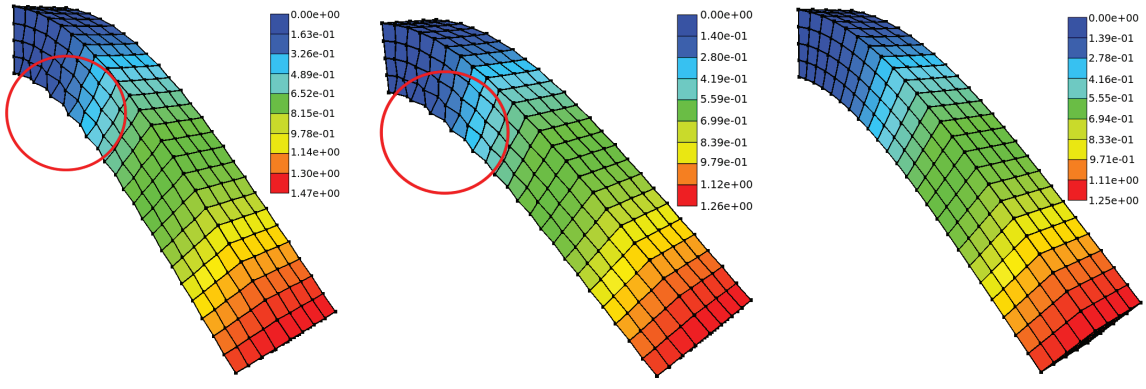
$$y(x) = \frac{\rho g l h}{24\ E\ I}(4Lx^3 - 6L^2x^2 - x^4), \tag{5.38}$$

taking into account the parameters: l, h - width and height of a beam, L - the length of the beam, ρ - the body density, gravity g and the moment of inertia I. Note that Euler-Bernouilli's solution is valid for a beam of bigger length than thickness and the equation represents the deflection of the neutral axis of a free-clamped beam.

It is visible with two simulation methods that the coarse curves are close to each other, but the fine meshes become much softer than the analytical solution, especially in case of the MSS-nFD model. In general both MSS experiments are not well aligned with the theoretical beam shape and even the new stiffness formulation used in MSS-nFD does not improve the behavior of the body enough to make up for the lack of face-diagonal springs.

On the other hand we can compare the results of MSS-CF with MSS-nFD for $E = 2$ MPa and $E = 10$ MPa presented in Fig. 5.18 and 5.19. MSS-nFD beam is much softer than MSS-CF and, even though the coarser beam is quite close to the analytical solution, we can see that with the change of mesh resolution the beam is flexed much more than the expected correct result. The MSS-nFD $8 \times 8 \times 48$ beam shape at the $E = 2$ MPa at the base may indicate the occurrence of the buckling instability that we talked about in Ch. 3.2.

Moreover, interesting property of the flexed beam shape can be observed in Fig. 5.15, where in the MSS-Baudet and MSS-nFD case there are apparent wrinkles at the compressed part of the beam (Fig. 5.15 (a) and (b)), which are absent when face diagonal springs and our stiffness formulation are used (Fig. 5.15 (c)). This inaccuracy (or even instability!) of MSS has both a visual and numerical impact on the simulated tissue. We can clearly see that by not using face-diagonal springs (even with our stiffness formulation and correction forces) we will encounter the wrinkles, which distort the results of a simulation. By using MSS-CF model we are able to avoid this kind of behavior and therefore provide visibly better and correct results.



(a) (b) (c)

FIGURE 5.15: Beam under gravity: (a) MSS-Baudet, (b) MSS-nFD, (c) MSS-CF. $E = 2$ MPa, $\nu = 0.25$, $g = -10$.

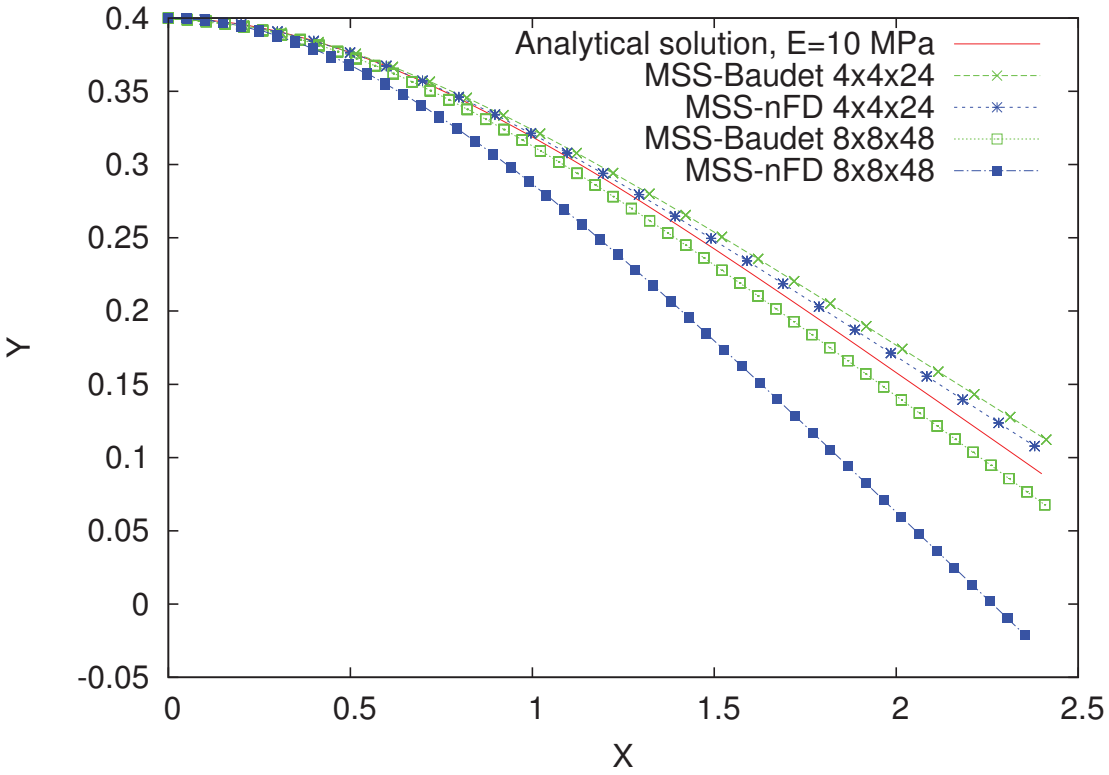


FIGURE 5.16: Beam under gravity: MSS-Baudet and MSS-nFD in comparison with analytical result. $E = 10$ MPa, $\nu = 0.25$, $g = -10$.

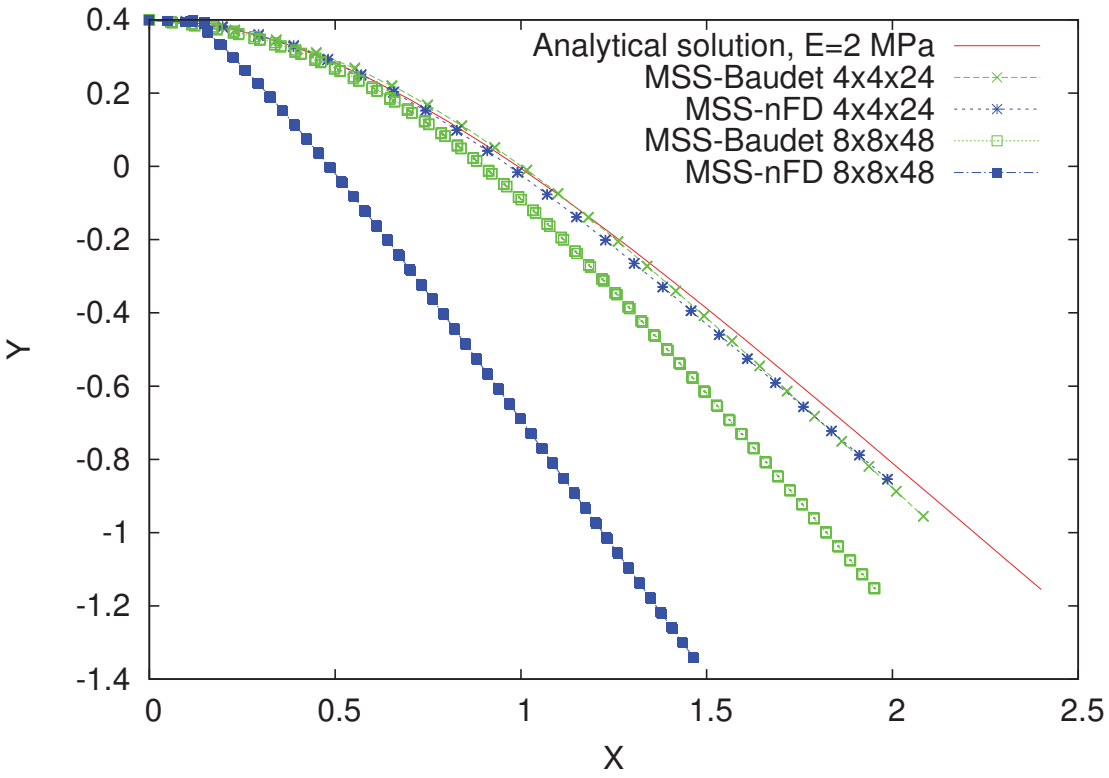


FIGURE 5.17: Beam under gravity: MSS-Baudet and MSS-nFD in comparison with analytical result. $E = 2$ MPa, $\nu = 0.25$, $g = -10$.

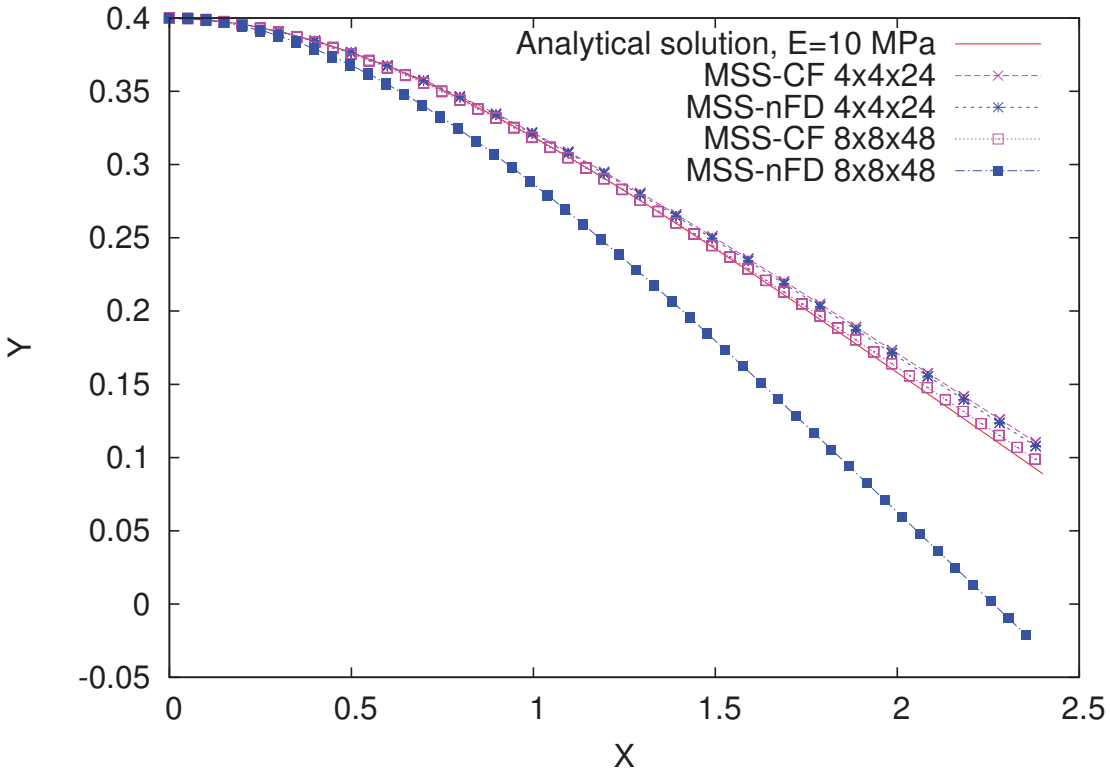


FIGURE 5.18: Beam under gravity comparison between MSS-CF, MSS-nFD and the analytical solution. $E = 10$ MPa, $\nu = 0.25$, $g = -10$.

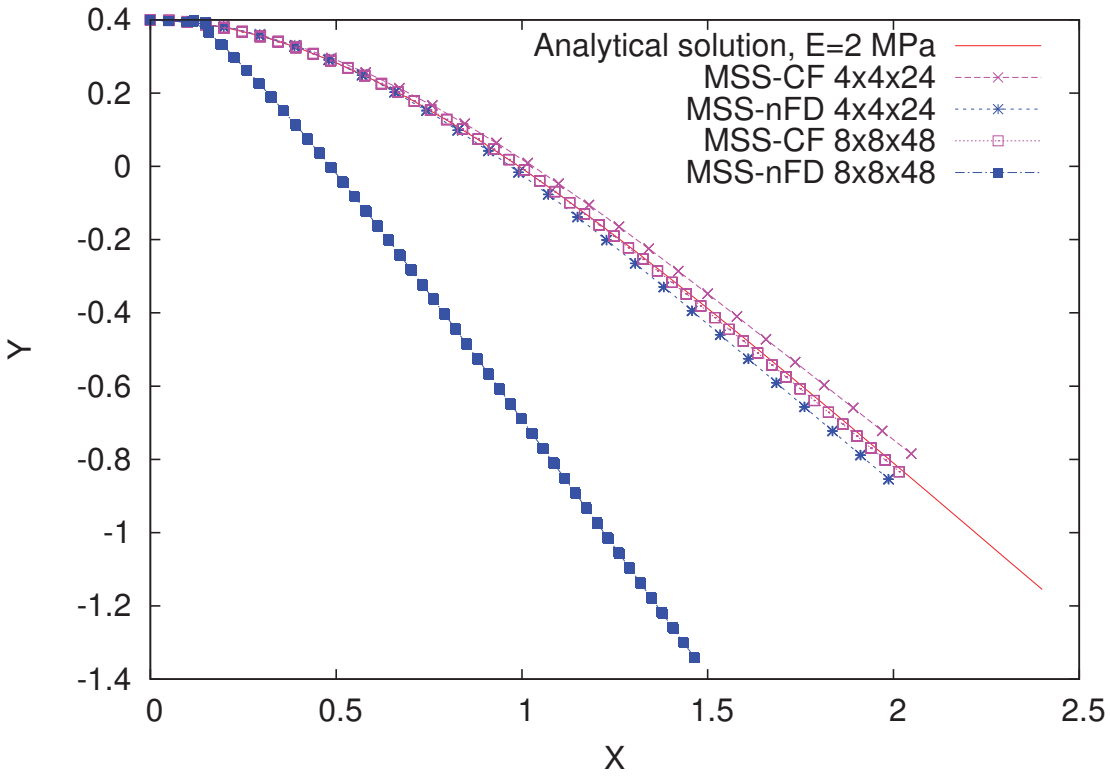


FIGURE 5.19: Beam under gravity comparison between MSS-CF, MSS-nFD and the analytical solution. $E = 2$ MPa, $\nu = 0.25$, $g = -10$.

Knowing the accuracy of the proposed Mass-Spring simulation techniques we can continue to more detailed part of the study, which includes the performance verification. In the following section we would like to show the potential of our solution and discuss the parallelization.

5.7 Performance Study

We studied the performance of all the MSS approaches using two types of CPU in the following configurations:

(I) Intel® Core™ i7 − 4790K @ 4.00 GHz; Ubuntu 14.04 LTS; PC;

(II) Intel® Xeon® E5 − 4607 0 @ 2.20 GHz; Ubuntu 14.04 LTS; virtual machine.

The first one is stationary computer running on the CPU (I) which has 4 cores, 2 threads each. The second part of experiments is run through a virtual machine (II) which is equipped in 6 cores, 2 threads. In the comparison we want to show the potential of our solution and verify how the current parallelization method works.

Second type of the processor is used to study the behavior of TBB parallelization, which we are able to run using 1, 2 or 3 CPUs, virtually having respectively 6, 12 and 18 cores. The time measurements are done using `time` unix function, which returns the real time (time of the parallel execution), user time (time of the serial execution) and system time (additional time spent by the system on communication *etc.*).

Another important note is the size of the samples. Table 5.5 below presents the exact dimensions of the meshes we used, together with the respective numbers of the volumes, vertices and springs. Due to the architecture of our implementation the number of springs for MSS is computed as a sum of the number of edge springs (shared by all the neighbouring volumes) and the number of inner diagonal springs, which are attributed to the 3-cells. The studied simulation methods include MSS-nFD, MSS-FD (Mass-Spring System with face-diagonal springs, but no correction forces) and MSS-CF. For MSS-FD and MSS-CF we have additional face diagonal springs (in comparison with MSS-nFD), thus we need to add 12 springs for each volume of the mesh, since the face-diagonal springs are the attributes of 2-cells. Note that the first four meshes are cubical and the last four are beams of a $1:1:6$ length ratio. We can see that the number of MSS-FD springs is roughly 2.5 times bigger than the number of MSS-nFD springs, therefore we can expect noticeably longer execution time from the MSS-FD method and even longer one in case of MSS-CF due to the additional computation of the force corrections for each volume.

Mesh size	**# of volumes**	**# of vertices**	# of springs	
			MSS-nFD	**MSS-FD**
$10 \times 10 \times 10$	1,000	1,331	7,630	19,630
$15 \times 15 \times 15$	3,375	4,096	25,020	65,520
$20 \times 20 \times 20$	8,000	9,261	58,460	154,460
$30 \times 30 \times 30$	27,000	29,791	194,490	518,490
$4 \times 4 \times 24$	384	625	3,136	7,744
$8 \times 8 \times 48$	3,072	3,969	23,232	60,096
$10 \times 10 \times 60$	6,000	7,381	44,680	116,680
$16 \times 16 \times 96$	24,576	28,033	178,816	473,728

TABLE 5.5: Details of the meshes used in the performance experiments.

The cause for using two types of meshes (cube and beam) is to show how the performance changes depending on the type of deformation. We decided to use cubically-shaped meshes for the same kind of displacement as presented in the compression experiments and the beam meshes for the beam under gravity test. The choice we made is dictated by the difference of the time execution when: (i) we apply only displacement as a source of the forces acting within the sample; (ii) there are external forces acting on the sample in a constant manner (in our case the gravity force) as the source of the internal forces.

The number of iterations and type of simulation are also important. We decided to execute $1{,}000$ iterations of the compression experiment and $300{,}000$ iterations of the beam flexion test. Moreover, as we already know from Ch. 5.4, the MSS-FD method does not compute the correction forces, since the Poisson ratio is equal to $1/4$ and consequently the κ parameter used to compute these forces is equal to 0. Thus, there is no need to compute the corrections, since for this value of Poisson's ratio the system is modeled correctly.

5.7.1 Performance Results

Figures 5.20 and 5.21 show the two experiments (compression and beam under gravity) performance plotted *versus* the number of mesh volumes. Of course the expected outcome of these simulations was that MSS-FD method will have slightly higher execution time than the simpler MSS-nFD, especially that the difference of the number of springs in each mesh is of $12 \times n$ order (the face-diagonal springs), where n is the number of volumes. Then, MSS-CF performed even worse, since the number of springs was already increased and it is necessary to compute the force corrections for each volume separately.

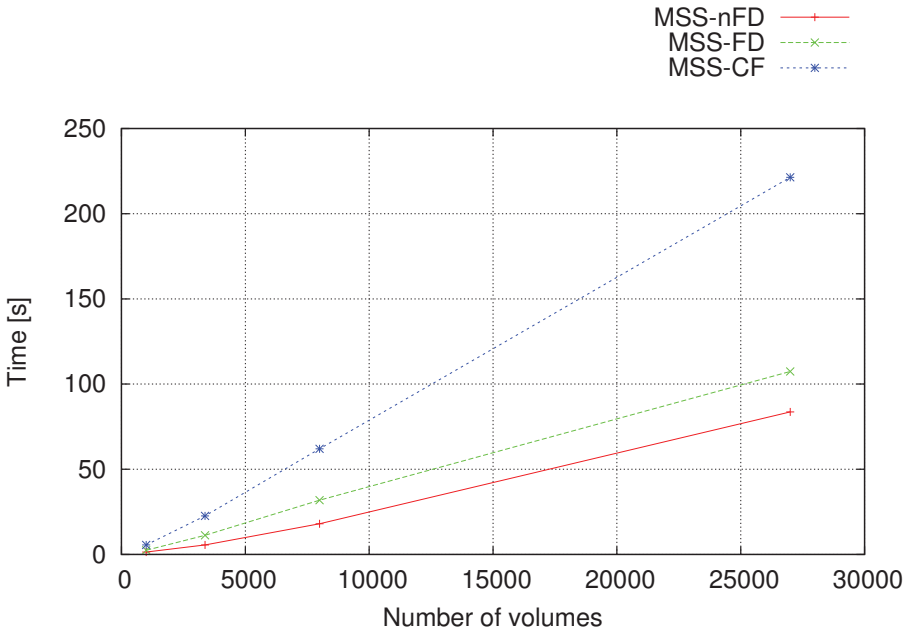


FIGURE 5.20: Real time performance of $1{,}000$ iterations *versus* the number of mesh elements (volumes) with CPU-software configuration I for the compression experiment and the simulation methods: MSS-nFD, MSS-FD, MSS-CF.

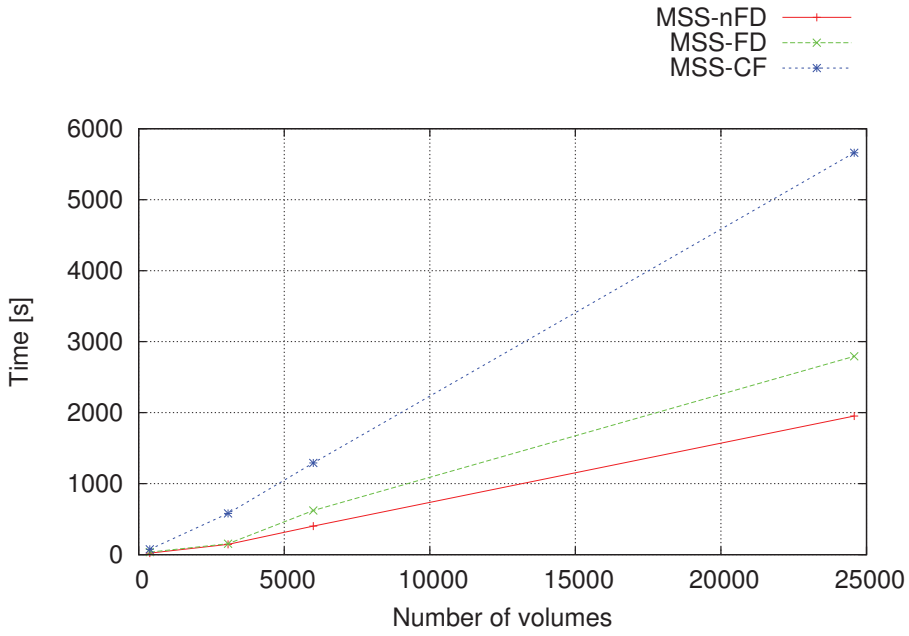


FIGURE 5.21: Real time execution of 300,000 iterations *versus* the number of mesh elements (volumes) on the CPU-software configuration I for the beam under gravity experiment and the three simulation methods: MSS-nFD, MSS-FD, MSS-CF.

5.7.2 Results of Our Parallel Simulations

To study the parallelization of all the methods on the beam and cube meshes we can compute the ratio of the user time t_u (*i.e.* the time of the parallel execution) to real execution time t_r (*i.e.* time of the serial execution) as a function of the number of mesh volumes. First, note that the beam mesh is used in the experiments of beam under gravity, which includes accumulation of external forces in addition to the spring forces, which is not necessary when using the cubical mesh used for the compression experiment. The t_u/t_r ratio in a perfect case shall give us the number of threads on which a simulation is performed. The corresponding curves for all the methods are presented in Fig. 5.22. Remembering that this particular configuration offers 8 virtual threads, we can see how good the parallelization method performs on the MSS-nFD and MSS-FD methods. However, since the MSS-CF requires additional computation time to iterate through all the volumes of the mesh, the t_u/t_r ratio rapidly drops in comparison to the two other methods. In fact it requires an improvement of the parallelization to be able to compute the correction forces for all the particles belonging to each volume without causing conflicts of the data access.

Now, let us study the t_u/t_r ratios of the experiments on the CPU-system configuration II. Since Fig. 5.22 clearly shown the dependency of the t_u/t_r ratio on the used method (*i.e.* for both experiments it is the MSS-CF method, which has visible drop down of the speedup in comparison to the other two methods), we can study only the compression experiment results. The curves on figures 5.23, 5.24 and 5.25 present the experiments performed on the machine with CPU-system configuration II, with sequentially 1, 2 and 3 CPUs. The number of virtual threads on each of those CPU numbers equals respectively 12, 24 and 36. Knowing that, we can assume that the ideal parallelization would take place when the t_u/t_r ratio would be exactly equal to the number of available virtual threads.

On Fig. 5.23 and 5.24 we can see the results of the compression experiment performed using MSS-nFD and MSS-FD methods. Indeed, we can observe that when only one CPU

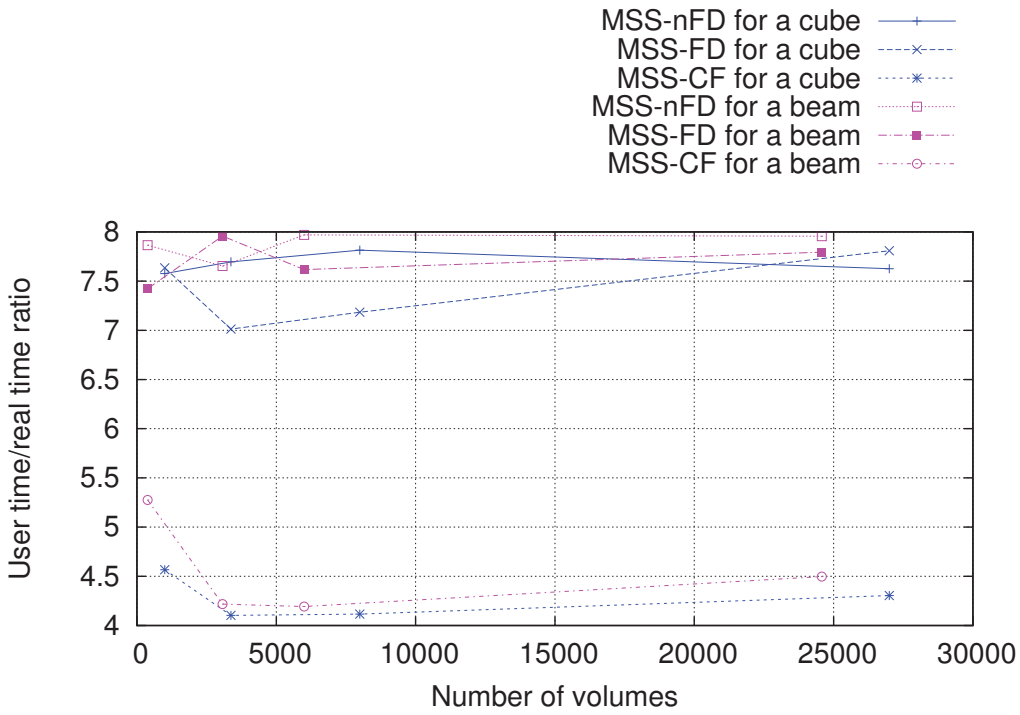


FIGURE 5.22: The t_u/t_r ratio *versus* the number of mesh volumes on the CPU-software configuration I - 8 threads.

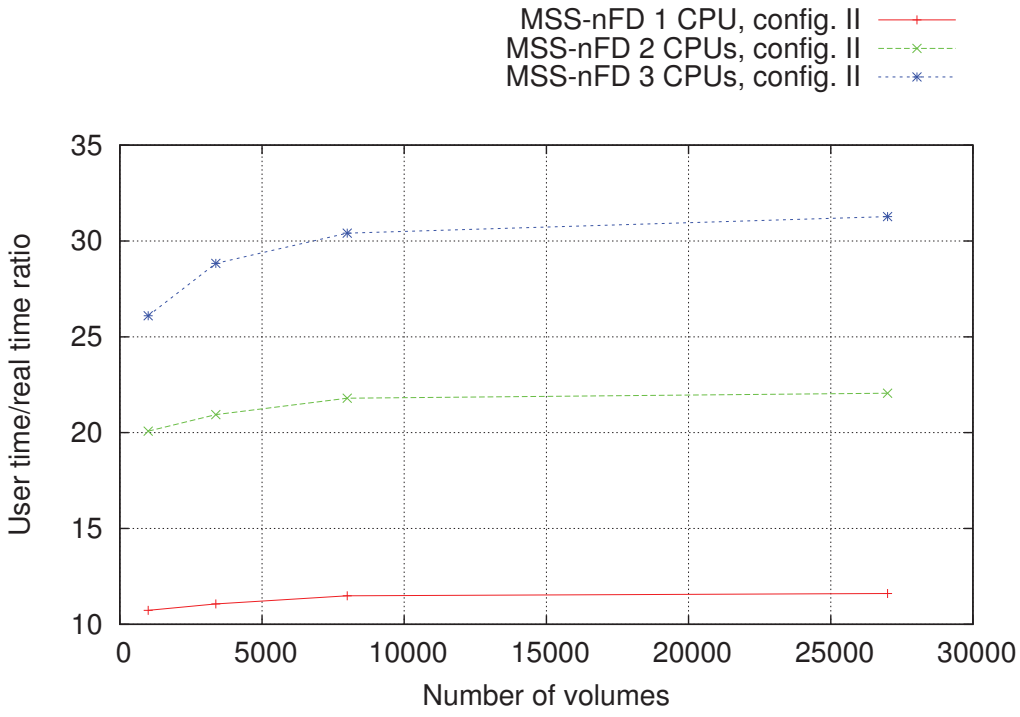


FIGURE 5.23: The t_u/t_r ratio *versus* the number of mesh volumes of the MSS-nFD method on the CPU-software configuration II.

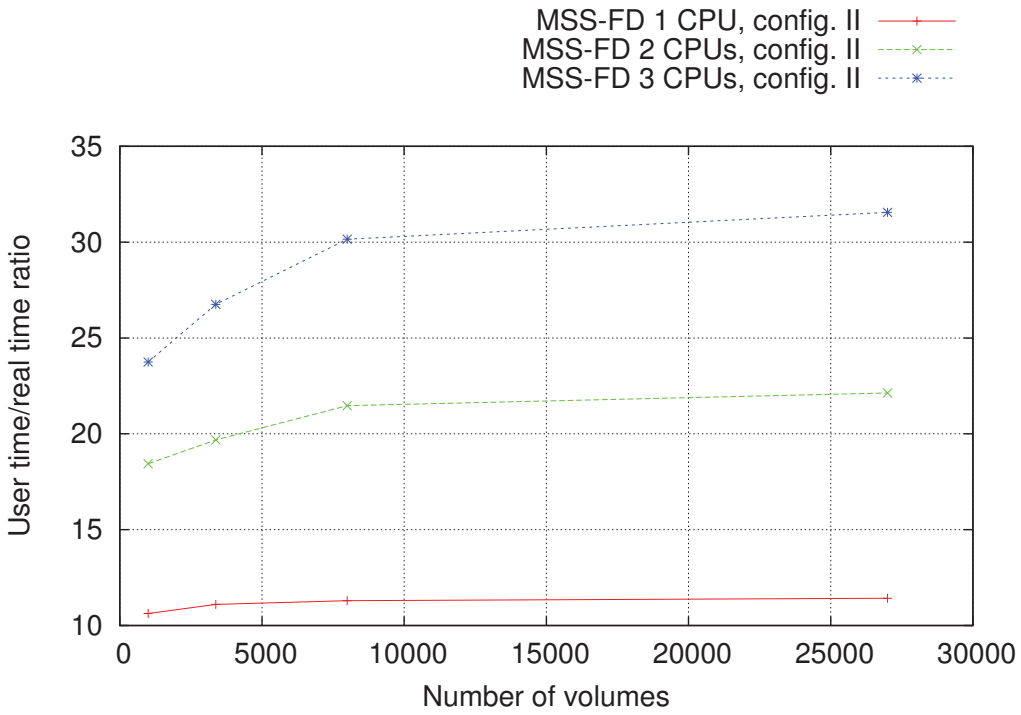


FIGURE 5.24: The t_u/t_r ratio *versus* the number of mesh volumes of the MSS-FD method on the CPU-software configuration II.

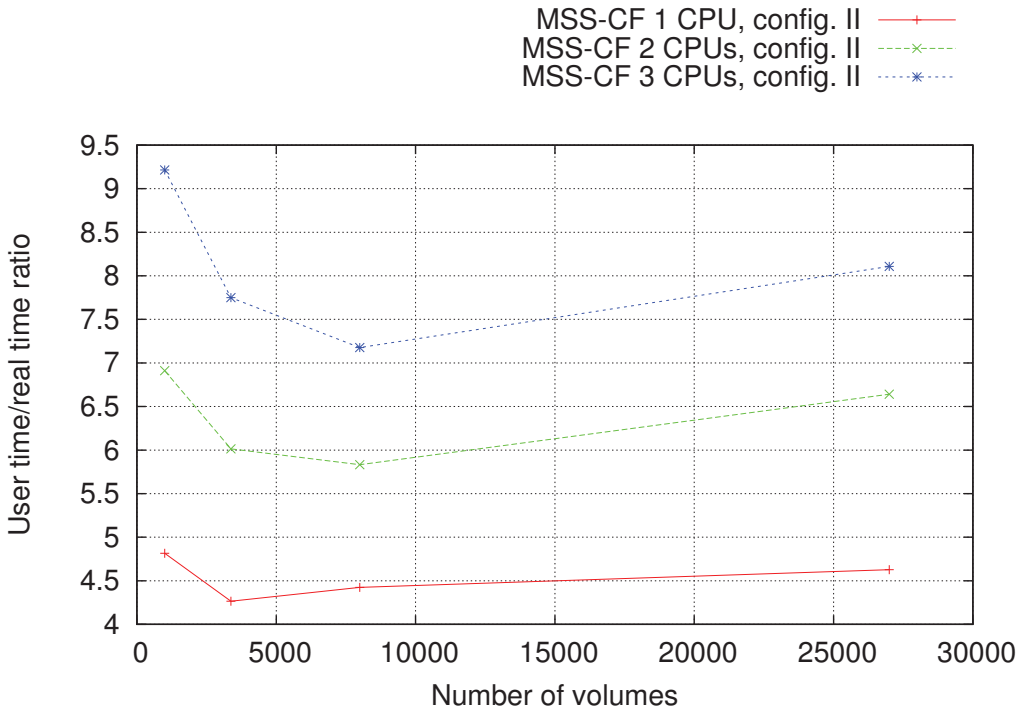


FIGURE 5.25: The t_u/t_r ratio *versus* the number of mesh volumes of the MSS-CF method on the CPU-software configuration II.

(12 virtual threads) is used the t_u/t_r ratio is the most advantageous in both cases. We obtained very good ratios also when two CPUs are used, but with the three CPUs we can see that the capabilities of the machines are not fully used. The third figure (Fig. 5.25) shows the results of the MSS-CF simulations and the first noticeable difference in comparison with the other simulation methods is the shape of the curves. Since the vast majority of the computation time is spent on the correction forces, the curve automatically becomes dependent on the number of volumes and rapidly drops when this number increases. We can see that the parallelization is the most advantageous for a smaller mesh and then evens out when the mesh becomes bigger than $\sim 8{,}000$ volumes due to the side and correction forces computations which are not parallelized.

5.8 Summary of the Results of the Experiments

To sum up, we have seen that the use of correction forces and all types of springs in Mass-Spring Systems plays a huge role in the body behavior and the shape of resulting deformations. Our aim is to propose a generic model, working regardless of the external forces and any topological modifications which a sample may undergo. The presented experimental results show that MSS-CF creates natural deformations of shapes, exactly as expected by the physical laws. Additionally, the numerical results indicate that it is possible to create an MSS model, which ensures the correct compressibility rate in a sample throughout all volumes (so at the local level) and not just as an average result (globally). These results encourage the use of mass-spring systems for simulation of real tissues, showing that the MSS-CF simulation is equally worth the attention from the point of view of accuracy and performance as the Finite Element Method. Even though we verified that the proposed MSS-CF method is able to perform a simulation slower than the presented MSS-nFD approach, we are sure to simulate deformations faster than the available FEM methods can. Additionally, we have shown that there is still room for the improvement of the performance of the MSS-CF approach, especially when iterating over the 3-cells of the mesh to calculate the correction forces.

5.9 Conclusions

In this chapter we introduced improvements to the classical MSS used to simulate isotropic, elastic tissues. The proposed contributions include dealing with the wrinkle instability and introducing physically-based stiffness formulations for all the used types of springs. The extension of the MSS to use all the real values of Poisson's ratio, namely the correction forces, is our final and major contribution. Working on cubical-symmetrical meshes, we have shown the effect of face-diagonal springs on the simulation; presented the new stiffness computation, which incorporates the mechanical information about a body into the model; and shown how to locally correct volumes in order to keep the simulation physically correct with adequate Poisson's ratio.

The proposed solutions offer stable, physically-justified basis with a simple implementation and many possibilities to improve and modify it. The easily adjustable parameters allow to successfully simulate compressible and incompressible isotropic materials (presented $\nu = 0.499$) with a minimal error. The solution is suitable for real-time applications (the mean performance of 10 second compression simulation of MSS-CF on a sample of $1{,}000$ elements on 8-thread processor is 5.579 s and the beam flexion of a 384 element mesh lasting 300 seconds can be simulated within 74.149 s on the same CPU), however

the additional computation of the volume and correction forces on every element obviously worsens the performance. It is possible to improve the computation time using TBB or consider other parallelization methods. What is more, the method has many conditions imposed on the model, which makes it complicated to adapt to other mesh structures.

Chapter 6

Mixed-Element Mesh for Mass-Spring System

In the physical simulation the context of topology and mesh structure overlaps with the accuracy of the representation. It is especially visible from the computer graphics point of view, since the biomedical field is focused more on the numerical methods, constitutive laws and the physical behavior of bodies. The requirement of accuracy is the main purpose of creating a model, which is simple and intuitive and in the same time allows a better representation of the simulated body with a performance better than this of a simple tetrahedral mesh.

The interest of using mixed-element meshes lies in a simulation of objects of various structure. Using mixed-elements one can create a mesh of the desired level of detail with the minimal number of volumes. Additionally, being able to perform remeshing of *e.g.* an all-hexahedral mesh to mixed-element mesh provides new capabilities also in topological modifications like cutting through volumes.

The methods to create mixed-element meshes are a few and the literature in this topic is limited, but our interest in the idea remains. Our attention is mainly drawn by the possibility of simulating detailed meshes with minimal number of elements, which improves even more the time of execution of a simulation. The main challenge to adapt a mixed-element mesh to MSS is to create a conformal mesh (which is currently not in our scope) and the definition of the acquisition of physical parameters. In this chapter we discuss about the mixed-element meshes as an open topic, since the contribution in creating a physically-correct mesh is our future purpose. We describe the structure of mixed element mesh, its implementation and especially the computation of all the necessary parameters to perform a physically correct simulation.

6.1 Mesh Structure and Shape Subdivision for MSS

We consider mixed element mesh having four different types of basic elements (see Fig. 6.1): (a) tetrahedron, which consists of 4 particles connected by 6 springs creating 4 triangular faces; (b) prism, which consists of 6 particles, 9 edge springs and 6 face-diagonal springs creating two parallel triangular and 3 rectangular faces; (c) pyramid, which consists of 5 particles, 8 edge springs and 2 face-diagonal springs creating a rectangular basis and 4 triangular faces; and (d) hexahedron which, as we already know from Ch. 3, consists of 8 particles, 12 edge springs, 12 face diagonal springs and 4 inner-diagonal springs creating 6 rectangular faces. Naturally, the lack of inner diagonal springs in prisms and pyramids can be explained by their architecture - to stabilize these elements more it would be necessary to *e.g.* add particle(s) in the middle of existing faces. These 4 types of elements are

generally enough to create a detailed mesh, where pyramids and prisms are used to join hexahedra and tetrahedra in an intuitive way.

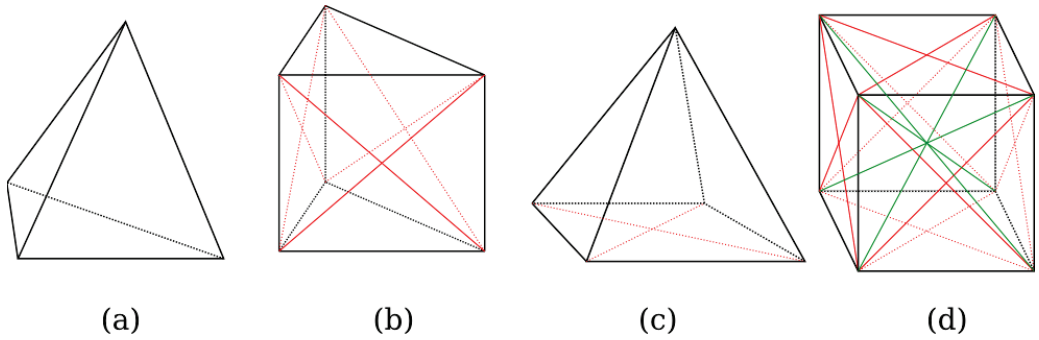


FIGURE 6.1: Basic MSS elements: (a) tetrahedron, (b) prism, (c) pyramid and (d) hexahedron. The rectangular faces are supported by face-diagonal springs (drawn in red); hexahedron is additionally supported with use of inner diagonal springs (drawn in green).

Moreover, each of the latter 3 elements can be easily subdivided into tetrahedra *e.g.* for the purpose of topological operations like cutting or refinement. Our interest in the shape subdivision lies in the possibility of filling an originally all-hexahedral mesh with the mixed-type elements. Such a mesh shall manifest exactly same physical behavior as its original version, which is easy to validate with our experimental setup.

The representation of the four basic mesh shapes plays a key role in establishing a correct LCC+MSS structure, as described in section 3.8.2. As we already saw in Fig. 6.1, to preserve the shape and ensure the correct behavior for shearing and tensile loads, the rectangular faces of prism, pyramid and hexahedron have to be supported by the face-diagonal springs. Additionally hexahedra are supported by the inner-diagonal springs, which are directly associated with the volumes.

Hexahedron Subdivision

Generally it is possible to subdivide a hexahedron in several ways. The possibilities of splitting a hexahedron result in different end scenarios with the following subdivisions:

- 5 tetrahedra;
- 2 prisms $\Rightarrow$ 1 prism $\rightarrow$ 3 tetrahedra $\Rightarrow$ 6 tetrahedra (Fig. 6.2 (a));
- 2 prisms $\Rightarrow$ 1 prism $\rightarrow$ 6 tetrahedra $\Rightarrow$ 12 tetrahedra (Fig. 6.2 (b));
- 6 pyramids $\Rightarrow$ 1 pyramid $\rightarrow$ 2 tetrahedra $\Rightarrow$ 12 tetrahedra.

5 tetrahedra.

First of the possibilities takes tetrahedra as the goal subdivision shapes. The objective is to split a hexahedron into 5 tetrahedra as shown in Fig. 6.3. Thus, a tetrahedron T_i, with $i \in \{1, ..., 5\}$ can be composed of the following points P_j, with $j \in \{1, ..., 8\}$, which originally construct a hexahedral element:

- $T_1 = \{P_1, P_2, P_4, P_6\}$,
- $T_2 = \{P_2, P_3, P_4, P_8\}$,

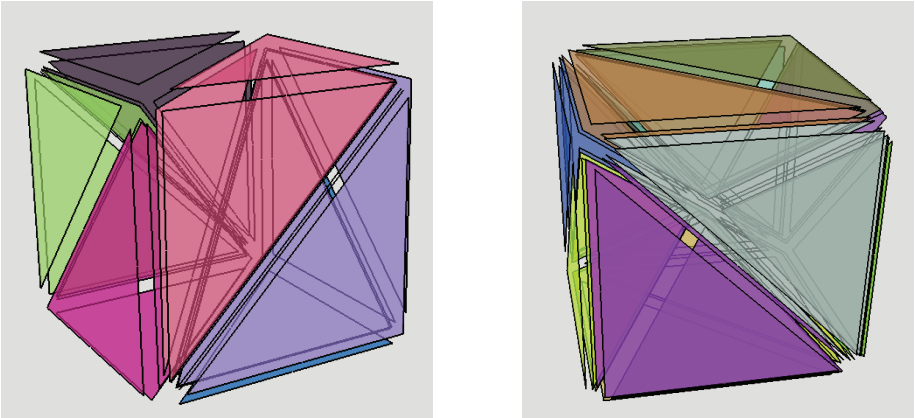


FIGURE 6.2: Hexahedron subdivision into (a) six tetrahedra or (b) twelve tetrahedra.

- $T_3 = \{P_2, P_5, P_6, P_8\}$,
- $T_4 = \{P_4, P_6, P_7, P_8\}$,
- $T_5 = \{P_2, P_4, P_6, P_8\}$.

This particular type of shape subdivision is usually not a preferred one in literature, mainly due to the lack of symmetry. Such a configuration is not invariant of the rotation, therefore we will not use it in this work.

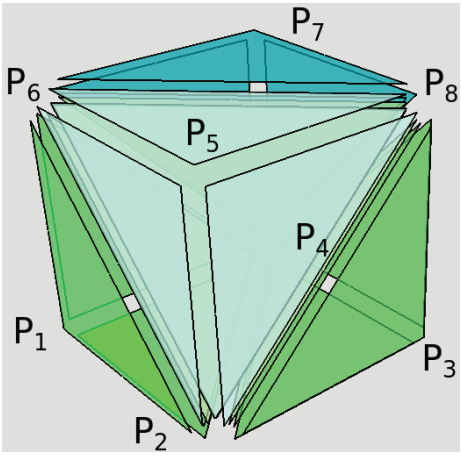


FIGURE 6.3: Hexahedron subdivision into 5 tetrahedra.

2 **prisms.**

The subdivision of a hexahedron can also be done using any of the two other basic shapes shown in Fig. 6.1. Thus, a hexahedral element can be split into two prisms if cut along the inner diagonal plane (Fig. 6.4).

This configuration has two further subdivision possibilities, as it is possible to split a prism in two ways. First of them is to subdivide it into three tetrahedra T_i, with $i \in \{1,\ 2,\ 3\}$ (see Fig. 6.5a). Thus, having 6 points which are used to build a prism: P_j, with $j \in \{1, ...,\ 6\}$ we have the following tetrahedra:

- $T_1 = \{P_1, P_2, P_3, P_5\}$,

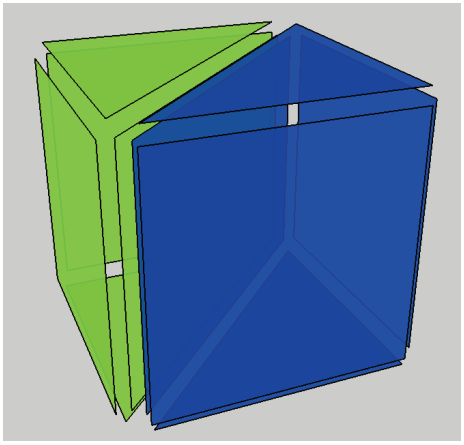
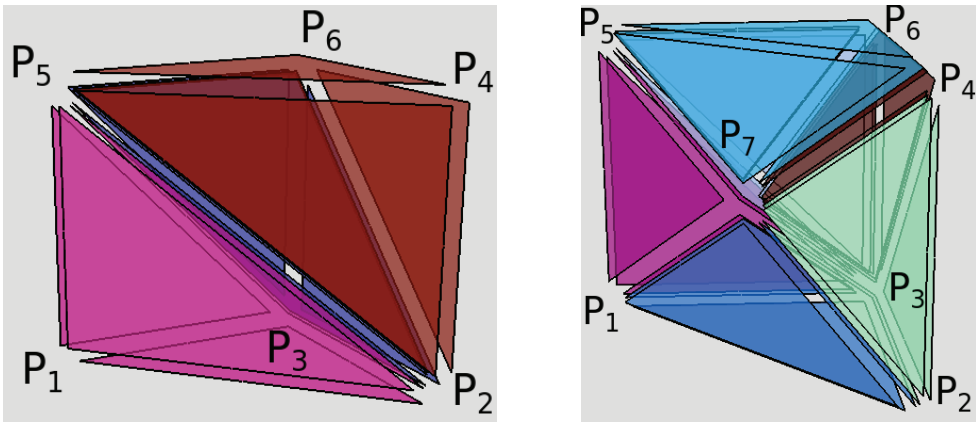


FIGURE 6.4: Hexahedron subdivision into two prisms.

- $T_2 = \{P_2, P_4, P_5, P_6\}$,
- $T_3 = \{P_2, P_3, P_5, P_6\}$.

Another possibility is to split a prism into 6 tetrahedral elements, with one additional mass point P_7 added in the middle of a rectangular face of the prism (see Fig. 6.5b). Then the prism shall consist of the following tetrahedral elements:

- $T_1 = \{P_1, P_3, P_5, P_7\}$,
- $T_2 = \{P_1, P_2, P_3, P_7\}$,
- $T_3 = \{P_2, P_3, P_4, P_7\}$,
- $T_4 = \{P_4, P_5, P_6, P_7\}$,
- $T_5 = \{P_3, P_4, P_6, P_7\}$,
- $T_6 = \{P_3, P_5, P_6, P_7\}$.



(a) 3 tetra subdivision (b) 6 tetra subdivision

FIGURE 6.5: Prism subdivision into (b) three tetrahedra or (b) six tetrahedra.

$6pyramids$**.** The third possibility of hexahedron subdivision is to split it into six pyramids. Each of the pyramids' base is one the hexahedron's face and the apex of each pyramid is in the hexahedron center, as shown in Fig. 6.6 (right).

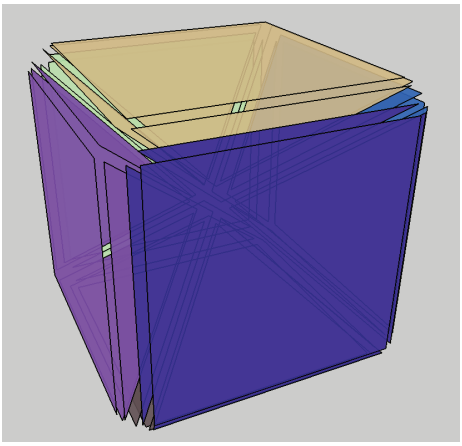


FIGURE 6.6: Hexahedron subdivision into six pyramids.

The subdivision of a pyramid into tetrahedra is the simplest, since it requires a single split along the diagonal of the base, which passes through the pyramid's apex, creating two tetrahedra, as depicted in Fig. 6.7. The created tetrahedral elements are composed of the following points:

- $T_1 = \{P_1, P_2, P_4, P_5\}$,
- $T_2 = \{P_2, P_3, P_4, P_5\}$.

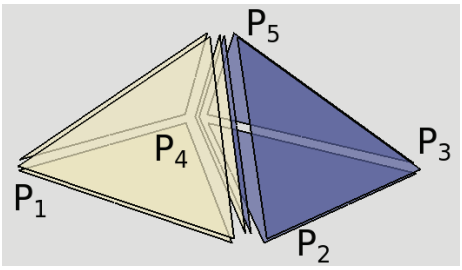
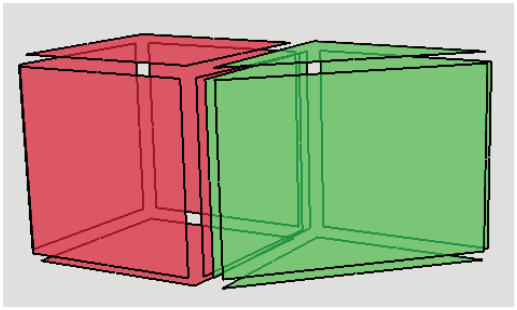


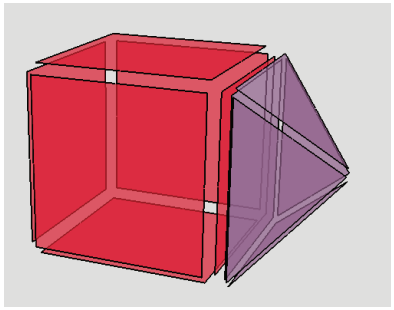
FIGURE 6.7: Pyramid subdivision into two tetrahedra.

6.1.1 Joining Mesh Elements

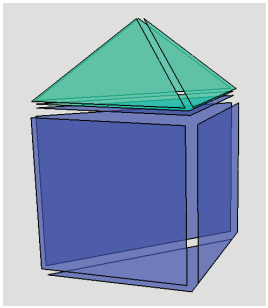
To create a mixed element mesh we can verify the possible configurations of the connections. As we already mentioned, the pyramids and prisms are the joint elements, which can be used to connect hexahedra with tetrahedra. The five basic connections are done using rectangular or triangular faces. This allows to connect a hexahedron with a pyramid (Fig. 6.8b) or a prism (Fig. 6.8a). A pyramid can be sewn with a prism in two ways: using matching rectangular (Fig. 6.8e) or triangular face. Finally, a tetrahedron, same as a hexahedral element, can be sewn with a prism (Fig. 6.8c) or a pyramid (Fig. 6.8d), but using their matching triangular faces.



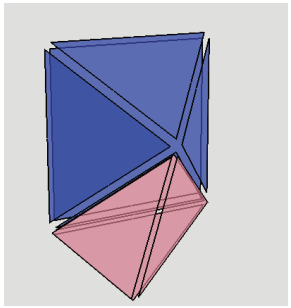
(a) Hexahedron + prism



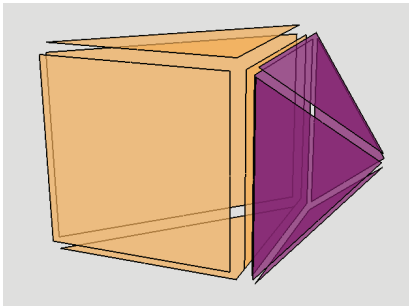
(b) Hexahedron + pyramid



(c) Tetrahedron + prism



(d) Tetrahedron + pyramid



(e) Prism + pyramid

FIGURE 6.8: Possible sewing scenarios.

6.1.2 Parameter Acquisition

After having the shape subdivision and possible element configurations established we can move to the acquisition of physical information. In MSS the most important parameters are mass of a vertex, volume of a 3D element and stiffness of a spring. As we already know the subdivision scheme of each of the basic shapes and we need the computation of volumes only as a pre-computational step (or an update after topological modifications) we can compute the exact volume in every case instead of calculating only an approximation.

Mass

The basic principle described by Eq. (3.19) remains intact: a particle is assigned a mass, which is a sum of masses derived from all the neighboring elements it belongs to. Previously we were interested only by a mass of a particle belonging to an all-hexahedral mesh. Now however, we would like to obtain an adequate mass of a particle, which belongs to several, possibly different 3D elements. The method to compute it remains the same, with the difference of computation of the mass of each basic element. The general formula to compute a mass m_v of any of the four basic elements, which is built with n particles is:

$$m_v = \frac{\rho\, V}{n}, \tag{6.1}$$

where ρ is the body density and V is the object's volume. Then, the formula to obtain a mass of a single particle m, which belongs to i 3D elements is computed by summing these masses for all the neighboring volumes i:

$$m = \sum_i m_v^i. \tag{6.2}$$

This generalization allows us to be generic and focus only on the computation of volumes of the 3D objects.

Volume

The computation of a volume of a 3D element is important not only to establish the mass of each particle of a MSS, but also as a potential tool to introduce volume correction forces into a mixed-element mesh MSS.

Tetrahedron. A volume of a regular tetrahedron is classically computed using the simple equation, which uses the length of tetrahedron's arm l_t: $V = l_t^3/(6\sqrt{2})$. However in our case, which aims at a generic treatment of the shapes, it has to be computed by a general formula, which is independent of the tetrahedron's arms' lengths:

$$V_{tetra} = \frac{1}{3} A_{base} H, \tag{6.3}$$

where A_{base} is the area of the base of a tetrahedron, meaning any of the face triangles, and H is the height of the tetrahedron measured from this base.

Other basic shapes. Using the formula of tetrahedron volume computation V_{tetra} we can simply establish the volumes of all the other basic shapes using the previously presented subdivision. Therefore, with a prism consisting of 3 tetrahedra, a pyramid consisting of 2 tetrahedra and a hexahedron consisting of 6 tetrahedra we have:

$$V_{prism} = \sum_{i=1}^{3} V_{tetra}^i \tag{6.4}$$

$$V_{pyramid} = \sum_{i=1}^{2} V_{tetra}^i \tag{6.5}$$

$$V_{hexa} = \sum_{i=1}^{6} V_{tetra}^i \tag{6.6}$$

Using this method we can easily compute the initial or updated volumes of the 3D shapes and use them to correctly compute the masses of the points they belong to. In the same time we are sure that the computation is generic *i.e.* each of the shapes after a deformation can still be subdivided into tetrahedra in the same way as the initial figure, so the computation is not affected by the shapes' geometry.

Spring Stiffness

The full discussion about the literature in the topic of obtaining stiffness for springs in MSS was undertaken in Ch. 3.4.3. From there we already know the variety of the methods used to compute or approximate this parameter. Because of its importance and the influence on the behavior of the simulated tissue we have presented our own idea in this matter in Ch. 5.1 for all-hexahedral symmetric isotropic meshes.

The mixed-element mesh is a much greater challenge though, since the variety of the shapes and our will to approach the topic with genericity do not allow us to use just any formulation. Additionally, since the mixed-element meshes were by now mainly used for FEM simulations, the computation of the spring stiffness was simply not needed. To fill in the gap in this topic we decided to approach the problem in a similar way as we have in Ch. 5.1 for the cubical elements. The full derivation of the stiffnesses of the mixed-element meshes is in our future work plan, but we want to stress out that it is the last step to obtain a full definition of this architecture for MSS. We aim at a formulation which takes into account the physical body parameters (defined by Young's modulus and Poisson's ratio), which involves the use of the face-diagonal and inner-diagonal springs and which allows simulation of real tissues with the same accuracy as our current model defined in the previously presented contributions. Adapting the mixed-element mesh to the idea of correction forces (see Ch. 5.2) is also one of our main goals.

6.2 Mixed-Element Mesh Implementation

Our work on the mixed-element meshes features the introduction of new basic shapes and their initialization (except of the spring stiffness parameters). We present the implementation details of the mixed-element mesh in TopoSim framework below.

To implement the mixed-element mesh shapes we need to add basic shapes to the lowest level of our implementation *i.e.* 3-map part (see Fig. 3.18). Since we are dealing with the 3-map elements we can work directly with the `Volume_LCC` class. All the basic shapes need to initialized in this class based on their c-maps. We can easily distinguish them by the number of vertices they are built with.

The next step is to associate the face-diagonal springs of a pyramid and a prism to the 3-cells, which means, that we need to look into the `Volume_LCC_MSS` class. Here we had implemented and associated the additional springs and introduced all the small '*get*' functions, which are used to differentiate between basic shapes and allow to use the generic functions (like the *add_force* function) without any adaptations. To do that we need to mind the distinction between different types of basic elements, since every one of them has a different number of face-diagonal (and inner diagonal in case of a hexahedron) springs. The list below presents the number of vertices and face-diagonal springs, which we need to take into account:

- tetrahedron - 4 vertices - 0 face-diagonal springs;
- pyramid - 5 vertices - 2 face-diagonal springs;
- prism - 6 vertices - 6 face-diagonal springs;
- hexahedron - 8 vertices - 12 face-diagonal springs.

The 'get' functions will then return the appropriate shape type, number of vertices, number of face-diagonal springs and the spring extremities.

The last function to be modified is the initialization of masses in the `Object_LCC_MSS` class: *initialize_masses*. As we have already explained in Ch. 6.1.2, the masses are computed thanks to a generic formulation, which uses the volume V of a 3-cell as one of the components. This means that we need to modify the initialization function to compute the exact volumes of our basic shapes using the previously defined shape subdivision into tetrahedra. Then, the function will return the resulting volume as a sum of the volumes of all the tetrahedra which compose particular basic shape.

6.3 Conclusions

Having a full mixed-element mesh formulation we will be able to simulate various real materials with a new level of accuracy. Additionally, the presented implementation, especially the most low-level part, which is building the LCC structure, can be used by different simulation techniques, being not only MSS but also MT or even FEM. Finally, it will be possible to perform a physically correct simulation with a mixed-element mesh using Mass-Spring System, which, to our knowledge, is not yet possible.

We have already verified that mixed-element meshes work well in our framework with the presented computation of mass and assigned fixed spring stiffness parameters. When the physically-based spring stiffness computation will be correctly derived, this contribution will open new door to research in the direction of mixed-element meshes used for MSS and hopefully will draw attention of researchers to work more on this topic not only from the point of view of FEM, but also other simulation methods.

Chapter 7

Conclusions and Perspectives

Context of the Thesis Subject

Modeling and simulation of behavior of deformable bodies, especially soft tissues, was the general subject of this thesis. Our first goal was to integrate the physical laws based on real experiments into our framework to model behavior of real bodies. The second, more general goal was to find a balance between the speed of calculations and precision. Because of our assumptions and objectives we have chosen to use Mass-Spring System, which, unlike continuous models, allows simple and fast calculations. However, due to its natural limitations and instabilities we had to improve this simulation technique to be able to reach our goals.

Contributions

In this thesis, we proposed several novel approaches to deal with MSS limitations and improve its accuracy in simulating real tissues. They are suitable for modeling elastic deformable bodies at small and large strains while controlling key behaviors specified by the physical body parameters.

- In the first part we presented a **non-linear force formulation** for the springs interactions, which is easily adaptable for simulation of real tissues at small and large deformations. It has proven to control the shape of the stress-time curve up to 100% strain with only 4 parameters. The method has the advantage of being easy to implement and reformulate if needed. Nonetheless, the formula needs an improvement to fit the curves based on the presented bi-power constitutive law and better adjust to the speed of body deformation.

- The second part of the thesis described **physically based formulation of stiffness computation** complemented with **force corrections** to allow use of full range of the Poisson ratio. This contribution deals well with global volumetric changes in a locally-oriented MSS. As we have presented in the experimental validation part, our hybrid approach is able to simulate the bodies from absolutely compressible ($\nu = 0$) to nearly incompressible ($\nu = 0.499$). Additionally, the presented formulation of stiffness parameters for three types op springs in an all-hexahedral mesh is not restricted to specific deformation types. The improvements stabilize the MSS and deal well with the natural MSS limitations and instabilities. The method needs however to be parallelized, since the time of execution of the hybrid model largely exceeds our target. What is more, the presented stiffness formulation works only for all-hexahedral meshes, mainly due to the assumption of isotropy we made.

- In the last part we have also presented the preliminary work concerning **mixed-element mesh** and its implementation. Our idea is to be able to create a mesh of a complex structure with minimal number of elements to allow fast and physically accurate simulation. We have successfully created and implemented the topological structure of this lattice. The mixed-element MSS mesh has also a good basis for the future implementation of volume correction forces thanks to the element subdivision. However, our work on the derivation of the stiffness parameters requires complex computations and a thorough validation, especially for the joint elements: pyramids and prisms.

We have implemented all of the above contributions in our framework - TopoSim. The methods are introduced in the framework to simulate the desired behaviors with robustness, speed and efficiency. We took into account the computational cost as well as the memory usage. The volume correction forces are adaptable to other physically-based simulation methods. Similarly, the mixed-element mesh structure can be easily extended to any modeling technique thanks to the presented choice of its topological structure.

Discussion

All the presented experimental results aim at a correct simulation of chosen body features, which we have validated in the previous chapters. Nonetheless, there are several particular features which can be discussed when it comes to this validation. Let us sum up the following characteristics of a successful physical simulation of deformable bodies:

- **Accuracy.** Non-linear formulation of forces presented in Chapter 4 controls well the shape of the curve, lining up with the analytical results. However the magnitude of each curve should strongly depend on the speed of deformation (strain rate), so the method still needs improvement in this matter. Additionally the method needs proper validation in various deformation scenarios *e.g.* compression or torsion, since the only validation we proposed by now is in shearing. On the other hand, the hybrid 3D improvement of MSS presented in Chapter 5 proves to be as accurate as a Finite Element Method in simulating elastic bodies. It has proven to accurately model the volumetric behavior of soft tissues under various deformations up to 20% strain, which is very satisfying for a linear deformation model.
- **Speed.** The non-linear force formulation is a simple solution to introduce non-linearity into MSS and it performs equally as the linear approach. Thus, the speed of simulation is naturally the same as for the linear MSS and was not studied in detail. The hybrid MSS however lacks parallelization in the computation of correction terms and therefore doubles the time of simulation execution.
- **Genericity.** All of the presented approaches were carefully evaluated to be as generic as our framework used for simulations. The non-linear force formulation can model any real organ with parameters defined with the bi-power constitutive law. In the same time, the hybrid MSS is not limited by the type of deformation to model. Indeed, this improvement can be used on any mesh element, since the formulation itself takes into account the volume change and physical body parameters, but not the mesh geometry.
- **Stability.** We have solved the problem of natural MSS wrinkle and buckling instabilities by using three types of springs in all-hexahedral meshes with the presented formulation of stiffness parameters.

The results of our experiments convince us that working on MSS structure can be fast, easy and, what is most important, give physically correct results. Another undeniable advantage of Mass-Spring System is the possibility to perform interactive simulations, thanks to the underlying LCC structure, which is not possible with the commonly used classical FEM. Despite the drawbacks of the chosen approaches we have shown that the choice of the simulation method in the biomedicine can be expanded to MSS, which provides new possibilities in the physical simulation field thanks to our improvements.

Future work

The MSS simulation method has been largely omitted in the field of medical modeling due to its lack of accuracy and supposed inability to introduce physically-based behaviors or deformable bodies. However our work have proven that MSS is still a plausible solution to model real organs fast and precisely. Nonetheless, there still remains a lot to work on in the future.

The first task we ought to focus on is the physically-based formulation of stiffness parameters for the elements of a mixed-type mesh. This work will significantly improve the performance of our simulations thanks to use of smaller number of mesh elements while keeping the same level of detail. To do that we want to follow the derivation method presented on hexahedral elements, however the simplifications done for hexahedra are not possible on other 3D elements *e.g.* due to the symmetry. Naturally, the mixed-type mesh formulation can easily be extended by our volume correction forces to control the volumetric changes with use of any natural value of the Poisson ratio.

Our method for simulating non-linear deformations with specially designed force formulation is able to control the shape of the curve, however lacks in accuracy to model the correct magnitude of deformation. We aim at improving the existing formulation to match the constitutive law at different strain rates, which currently is the biggest drawback of our formulation. Additionally we want to be able to model the behavior of various bodies (performing verification of different deformations) using all the parameters which define a body in case of the bi-power law.

The most interesting step to work on will be our hybrid MSS extension to model large deformations. In fact the volume correction forces are supposedly suitable for large deformations, which needs a thorough validation with suitable, improved formulation of non-linear computation of forces. The combination of the two methods shall prove MSS to accurately model non-linear deformations of real tissues with possibility of adjusting their compressibility. Such a formulation would be useful in combination with haptic devices to model real organs in real time with the possibility of topological modifications.

The remaining task will involve many experiments done on different types of elements and model geometries to verify how to model real behavior of soft tissues best. Additionally we will need to compare a dynamic FEM and MSS simulations to verify the stability, accuracy and performance of the two approaches. Once we establish that, we get a simulation environment which will allow us to model different types of behavior of various tissues while being able to perform topological modifications during a simulation. We take into account possible modifications to be done to ensure the stability of physical models when changing the mesh topology.

Bibliography

[1] *About Society for Simulation in Healthcare*. 2017. URL: http://www.ssih.org/About-SSH (visited on 05/16/2017).

[2] Awais Ahmed. "Extended finite element method (XFEM)-modeling arbitrary discontinuities and failure analysis". In: *Research degree thesis* (2009).

[3] Jérémie Allard, Stéphane Cotin, François Faure, Pierre-Jean Bensoussan, François Poyer, Christian Duriez, Hervé Delingette, and Laurent Grisoni. "SOFA - an Open Source Framework for Medical Simulation". In: *MMVR 15 - Medicine Meets Virtual Reality*. Vol. 125. Studies in Health Technology and Informatics. Palm Beach, United States: IOP Press, Feb. 2007, pp. 13–18.

[4] S. Arnab and V. Raja. "Chapter 4: Simulating a Deformable Object Using a Surface Mass Spring System". In: *2008 3rd International Conference on Geometric Modeling and Imaging*. July 2008, pp. 21–26. DOI: 10.1109/GMAI.2008.24.

[5] Fred S. Azar, Dimitris N. Metaxas, and Mitchell D. Schnall. "A Deformable Finite Element Model of the Breast for Predicting Mechanical Deformations under External Perturbations". In: *Academic Radiology* 8.10 (2001), pp. 965 –975. ISSN: 1076-6332. DOI: https://doi.org/10.1016/S1076-6332(03)80640-2.

[6] David Baraff and Andrew Witkin. "Large Steps in Cloth Simulation". In: *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '98. New York, NY, USA: ACM, 1998, pp. 43–54. ISBN: 0-89791-999-8. DOI: 10.1145/280814.280821.

[7] Ehsan Basafa, Farzam Farahmand, and Gholamreza Vossoughi. "A non-linear mass-spring model for more realistic and efficient simulation of soft tissues surgery". In: *Studies in health technology and informatics* 132 (2008), p. 23.

[8] Cagatay Basdogan. "Real-time simulation of dynamically deformable finite element models using modal analysis and spectral lanczos decomposition methods". In: (2000).

[9] K.J. Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice Hall, New Jersey, 1982.

[10] Klaus-Jürgen Bathe. "Conserving energy and momentum in nonlinear dynamics: a simple implicit time integration scheme". In: *Computers & structures* 85.7 (2007), pp. 437–445.

[11] Lukas M. Batteau, Alan Liu, J. B. Antoine Maintz, Yogendra Bhasin, and Mark W. Bowyer. "A Study on the Perception of Haptics in Surgical Simulation". In: *Medical Simulation: International Symposium, ISMS 2004, Cambridge, MA, USA, June 17-18, 2004. Proceedings*. Ed. by Stéphane Cotin and Dimitris Metaxas. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 185–192. ISBN: 978-3-540-25968-8. DOI: 10.1007/978-3-540-25968-8_21.

[12] V. Baudet, M. Beuve, F. Jaillet, B. Shariat, and F. Zara. "Integrating Tensile Parameters in Hexahedral Mass-Spring System for Simulation". In: *WSCG'2009*. 2009. ISBN: 9788086943930.

[13] T. Belytschko and T. Black. "Elastic crack growth in finite elements with minimal remeshing". In: *International Journal for Numerical Methods in Engineering* 45.5 (1999), pp. 601–620. ISSN: 1097-0207. DOI: 10.1002/(SICI)1097-0207(19990620)45:5<601::AID-NME598>3.0.CO;2-S.

[14] Ted Belytschko and Lawrence W. Glaum. "Applications of higher order corotational stretch theories to nonlinear finite element analysis". In: *Computers & Structures* 10.1 (1979), pp. 175 –182. ISSN: 0045-7949. DOI: https://doi.org/10.1016/0045-7949(79)90085-3.

[15] Jan Bender, Matthias Müller, Miguel A. Otaduy, and Matthias Teschner. "Position-based Methods for the Simulation of Solid Objects in Computer Graphics". In: *EUROGRAPHICS 2013 State of the Art Reports*. Girona, Spain: Eurographics Association, 2013.

[16] Jan Bender, Matthias Müller, Miguel A. Otaduy, and Matthias Teschner. "Position-based Methods for the Simulation of Solid Objects in Computer Graphics". In: *EUROGRAPHICS 2013 State of the Art Reports*. Girona, Spain: Eurographics Association, 2013.

[17] J. Berkley, S. Weghorst, H. Gladstone, G. Raugi, D. Berg, and M. Ganter. "Banded matrix approach to Finite Element modelling for soft tissue simulation". In: *Virtual Reality* 4.3 (Sept. 1999), pp. 203–212. ISSN: 1434-9957. DOI: 10.1007/BF01418156.

[18] Kiran S. Bhat, Christopher D. Twigg, Jessica K. Hodgins, Pradeep K. Khosla, Zoran Popović, and Steven M. Seitz. "Estimating Cloth Simulation Parameters from Video". In: *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA '03. San Diego, California: Eurographics Association, 2003, pp. 37–51. ISBN: 1-58113-659-5.

[19] G. Bianchi, B. Solenthaler, G. Székely, and M. Harders. "Simultaneous Topology and Stiffness Identification for Mass-Spring Models based on FEM Reference Deformations". In: *MICCAI 2004*. Vol. 2. Springer, Nov. 2004, pp. 293–301.

[20] Gérald Bianchi, Matthias Harders, and Gábor Székely. "Mesh Topology Identification for Mass-Spring Models". In: *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2003: 6th International Conference, Montréal, Canada, November 15-18, 2003. Proceedings*. Ed. by Randy E. Ellis and Terry M. Peters. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 50–58. ISBN: 978-3-540-39899-8. DOI: 10.1007/978-3-540-39899-8_7.

[21] D. Bielser and M. H. Gross. "Interactive simulation of surgical cuts". In: *Proceedings the Eighth Pacific Conference on Computer Graphics and Applications*. 2000, pp. 116–442. DOI: 10.1109/PCCGA.2000.883933.

[22] Richard Evelyn Donohue Bishop and DC Johnson. *The Mechanics of Vibration*. Cambridge University Press, 1979.

[23] F. Boosman. "Macintosh Windows". In: *Computer Gaming World* 32 (Nov. 1986), pp. 15, 32.

[24] David Bourguignon and Marie-Paule Cani. "Controlling Anisotropy in Mass-Spring Systems". In: *Computer Animation and Simulation 2000: Proceedings of the Eurographics Workshop in Interlaken, Switzerland, August 21–22, 2000*. Vienna: Springer Vienna, 2000, pp. 113–123. ISBN: 978-3-7091-6344-3.

[25] R. Bridson, R. Fedkiw, and J. Anderson. "Robust Treatment of Collisions, Contact and Friction for Cloth Animation". In: *ACM Trans. Graph.* 21.3 (July 2002), pp. 594–603. ISSN: 0730-0301.

[26] Morten Bro-Nielsen. "Modelling Elasticity in Solids Using Active Cubes — Application to Simulated Operations". In: *Computer Vision, Virtual Reality and Robotics in Medicine: First International Conference, CVRMed '95, Nice, France, April 3–6, 1995 Proceedings*. Ed. by Nicholas Ayache. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 535–541. ISBN: 978-3-540-49197-2. DOI: 10.1007/978-3-540-49197-2_70.

[27] Morten Bro-Nielsen and Stephane Cotin. "Real-time Volumetric Deformable Models for Surgery Simulation using Finite Elements and Condensation". In: *Computer Graphics Forum* 15.3 (1996), pp. 57–66. ISSN: 1467-8659. DOI: 10.1111/1467-8659.1530057.

[28] Joel Brown, Stephen Sorkin, Jean-Claude Latombe, Kevin Montgomery, and Michael Stephanides. "Algorithmic tools for real-time microsurgery simulation". In: *Medical image analysis* 6.3 (2002), pp. 289–300.

[29] David T. Chen and David Zeltzer. "Pump It Up: Computer Animation of a Biomechanically Based Model of Muscle Using the Finite Element Method". In: *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '92. New York, NY, USA: ACM, 1992, pp. 89–98. ISBN: 0-89791-479-1. DOI: 10.1145/133994.134016.

[30] F. Chen, L. Gu, P. Huang, J. Zhang, and J. Xu. "Soft Tissue Modeling using Nonlinear Mass Spring and Simplified Medial Representation". In: *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Aug. 2007, pp. 5083–5086.

[31] Kup-Sze Choi, Hanqiu Sun, and Pheng-Ann Heng. "An efficient and scalable deformable model for virtual reality-based medical applications". In: *Artificial Intelligence in Medicine* 32.1 (2004). Atificial Intelligence in Medicine in China, pp. 51 –69. ISSN: 0933-3657. DOI: https://doi.org/10.1016/j.artmed.2004.01.013.

[32] Gerard M Cooney, Kevin M Moerman, Michael Takaza, Des C Winter, and Ciaran K Simms. "Uniaxial and biaxial mechanical properties of porcine linea alba." In: *Journal of the mechanical behavior of biomedical materials* 41C (Oct. 2014), pp. 68–82. ISSN: 1878-0180. DOI: 10.1016/j.jmbbm.2014.09.026.

[33] S. Cotin, H. Delingette, and N. Ayache. "A Hybrid Elastic Model allowing Real-Time Cutting, Deformations and Force-Feedback for Surgery Training and Simulation". Anglais. In: *Visual Computer* 16.8 (2000), pp. 437–452. DOI: 10.1007/PL00007215.

[34] S. Cotin, H. Delingette, and N. Ayache. "Real-time elastic deformations of soft tissues for surgery simulation". In: *IEEE Transactions on Visualization and Computer Graphics* 5.1 (1999), pp. 62–73.

[35] Stéphane Cotin, Hervé Delingette, and Nicholas Ayache. "Real time volumetric deformable models for surgery simulation". In: *Visualization in Biomedical Computing: 4th International Conference, VBC'96 Hamburg, Germamy, September 22–25, 1996 Proceedings*. Ed. by Karl Heinz Höhne and Ron Kikinis. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 535–540. ISBN: 978-3-540-70739-4. DOI: 10.1007/BFb0046995.

[36] S. A. Cover, N. F. Ezquerra, J. F. O'Brien, R. Rowe, T. Gadacz, and E. Palm. "Interactively deformable models for surgery simulation". In: *IEEE Computer Graphics and Applications* 13.6 (Nov. 1993), pp. 68–75. ISSN: 0272-1716. DOI: 10.1109/38.252559.

[37] Elias Cueto, Siamak Niroomandi, David Gonzalez, Iciar Alfaro, Felipe Bordeu, Adrien Leygue, and Francisco Chinesta. "Real Time Simulation of Non-Linear Solids by PGD Techniques". In: *Material Forming ESAFORM 2012*. Vol. 504. Key Engineering Materials. Trans Tech Publications, Apr. 2012, pp. 467–472. DOI: 10.4028/www.scientific.net/KEM.504-506.467.

[38] Tong Cui, Aiguo Song, and Juan Wu. "Simulation of a mass-spring model for global deformation". In: *Frontiers of Electrical and Electronic Engineering in China* 4.1 (2009), pp. 78–82. ISSN: 1673-3584. DOI: 10.1007/s11460-009-0001-6.

[39] Benoît Dagon, Charles Baur, and Vincent Bettschart. "Real-time update of 3D deformable models for computer aided liver surgery". In: *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE. 2008, pp. 1–4.

[40] G. Damiand and P. Lienhardt. *Combinatorial Maps: Efficient Data Structures for Computer Graphics and Image Processing*. A K Peters/CRC Press, Sept. 2014.

[41] Guillaume Damiand. "Combinatorial Maps". In: *CGAL User and Reference Manual*. 4.10. CGAL Editorial Board, 2017. URL: http://doc.cgal.org/4.10/Manual/packages.html#PkgCombinatorialMapsSummary.

[42] B Fraeijs De Veubeke. "The dynamics of flexible bodies". In: *International Journal of Engineering Science* 14.10 (1976), pp. 895–913.

[43] G. Debunne, M. Desbrun, M. P. Cani, and A. Barr. "Adaptive simulation of soft bodies in real-time". In: *Proceedings Computer Animation 2000*. 2000, pp. 15–20. DOI: 10.1109/CA.2000.889022.

[44] S DeŁ, J Kim, and MA Srinivasan. "Virtual surgery simulation using a collocation-based method of finite spheres". In: (2001).

[45] H. Delingette. "Toward realistic soft-tissue modeling in medical simulation". In: *Proceedings of the IEEE* 86.3 (Mar. 1998), pp. 512–523. ISSN: 0018-9219. DOI: 10.1109/5.662876.

[46] Hilmi Demiray. "A note on the elasticity of soft biological tissues". In: *Journal of Biomechanics* 5.3 (1972), pp. 309 –311. ISSN: 0021-9290. DOI: http://dx.doi.org/10.1016/0021-9290(72)90047-4.

[47] Oliver Deussen, Leif Kobbelt, and Peter Tücke. "Using Simulated Annealing to Obtain Good Nodal Approximations of Deformable Bodies". In: *Computer Animation and Simulation '95: Proceedings of the Eurographics Workshop in Maastricht, The Netherlands, September 2–3, 1995*. Ed. by Demetri Terzopoulos and Daniel Thalmann. Vienna: Springer Vienna, 1995, pp. 30–43. ISBN: 978-3-7091-9435-5. DOI: 10.1007/978-3-7091-9435-5_3.

[48] Donald L. Dewhirst and Peter M. Grinsell. "Joining Tetrahedra to Hexahedra". In: *MSC 1993 World Users' Conference Proceedings*. Arlington, Virginia, May 1993.

[49] Donald L. Dewhirst, Sreedhar Vangavolu, and Harley Wattrick. "The Combination of Hexahedral and Tetrahedral Meshing Algorithms". In: *4th International Meshing Roundtable*. Sandia National Laboratories, Oct. 1995, pp. 291–304.

[50] R. Diziol, J. Bender, and D. Bayer. "Robust Real-time Deformation of Incompressible Surface Meshes". In: *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA '11. Vancouver, British Columbia, Canada: ACM, 2011, pp. 237–246. ISBN: 978-1-4503-0923-3.

[51] Y. Duan, W. Huang, H. Chang, K. K. Toe, T. Yang, J. Zhou, J. Liu, S. K. Teo, C. W. Lim, Y. Su, C. K. Chui, and S. Chang. "Synchronous simulation for deformation of liver and gallbladder with stretch and compression compensation". In: *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. July 2013, pp. 4941–4944. DOI: 10.1109/EMBC.2013.6610656.

[52] Y. Duan, W. Huang, H. Chang, W. Chen, J. Zhou, S. K. Teo, Y. Su, C. Chui, and S. Chang. "Volume Preserved Mass-spring Model with Novel Constraints for Soft Tissue Deformation." In: *IEEE journal of biomedical and health informatics* 2194.c (Nov. 2014), pp. 1–12. ISSN: 2168-2208. DOI: 10.1109/JBHI.2014.2370059.

[53] Raed E El-Jawahri, Tony R Laituri, Jesse S Ruan, Stephen W Rouhana, and Saeed D Barbat. "Development and validation of age-dependent FE human models of a mid-sized male thorax". In: *Stapp car crash journal* 54 (2010), p. 407.

[54] Olaf Etzmuß, Michael Keckeisen, and Wolfgang Straßer. "A fast finite element solution for cloth modelling". In: *Computer Graphics and Applications, 2003. Proceedings. 11th Pacific Conference on*. IEEE. 2003, pp. 244–251.

[55] E. Flechon, F. Zara, G. Damiand, and F. Jaillet. "A generic topological framework for physical simulation". en. In: *WSCG'2013*. June 2013, pp. 104–113. ISBN: 978-80-86943-74-9.

[56] E. Flechon, F. Zara, G. Damiand, and F. Jaillet. "A unified topological-physical model for adaptive refinement ". en. In: *VRIPHYS'2014*. Sept. 2014, pp. 39–48. DOI: 10.2312/vriphys.20141222.

[57] A Frisoli, LF Borelli, C Stasi, M Bellini, C Bianchi, E Ruffaldi, G Di Pietro, and M Bergamasco. "Simulation of real-time deformable soft tissues for computer assisted surgery". In: *The International Journal of Medical Robotics and Computer Assisted Surgery* 1.1 (2004), pp. 107–113.

[58] Thomas Fritz, Oussama Jarrousse, and Olaf Dössel. "Adapting a Mass-Spring system to energy density function describing myocardial mechanics". In: *4th European Conference of the International Federation for Medical and Biological Engineering: ECIFMBE 2008 23–27 November 2008 Antwerp, Belgium*. Ed. by Jos Vander Sloten, Pascal Verdonck, Marc Nyssen, and Jens Haueisen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 2003–2006. ISBN: 978-3-540-89208-3. DOI: 10.1007/978-3-540-89208-3_477.

[59] YC Fung. "Biomechanics: mechanical properties of living tissues". In: (1981).

[60] Y.C. Fung. *Biomechanics: Mechanical Properties of Living Tissues*. Biomechanics. Springer New York, 1993. ISBN: 9780387979472.

[61] YC Fung. "Biorheology of soft tissues". In: *Biorheology* 10.2 (1973), pp. 139–155.

[62] YC Fung. "Elasticity of soft tissues in simple elongation". In: *American Journal of Physiology–Legacy Content* 213.6 (1967), pp. 1532–1544.

[63] Fabio Ganovelli, Paolo Cignoni, Claudio Montani, and Roberto Scopigno. "A Multiresolution Model for Soft Objects Supporting Interactive Cuts and Lacerations". In: *Computer Graphics Forum* 19.3 (2000), pp. 271–281. ISSN: 1467-8659. DOI: 10.1111/1467-8659.00419.

[64] Allen Van Gelder. "Approximate Simulation of Elastic Membranes by Triangulated Spring Meshes". In: *Journal of Graphics Tools* 3.2 (1998), pp. 21–41. DOI: 10.1080/10867651.1998.10487490.

[65] Sarah F F Gibson and Brian Mirtich. "A Survey of Deformable Modeling in Computer Graphics". In: *Mitsubishi Electric Information Technology Center America* (Nov. 1997).

[66] François Goulette and Zhuo-Wei Chen. "Fast computation of soft tissue deformations in real-time simulation with Hyper-Elastic Mass Links". In: *Computer Methods in Applied Mechanics and Engineering* 295 (2015), pp. 18 –38. ISSN: 0045-7825. DOI: http://dx.doi.org/10.1016/j.cma.2015.06.015.

[67] XiaoHu Guo and Hong Qin. "Meshless methods for physics-based modeling and simulation of deformable models". In: *Science in China Series F: Information Sciences* 52.3 (2009), pp. 401–417. ISSN: 1862-2836. DOI: 10.1007/s11432-009-0069-x.

[68] Michael Hauth, Olaf Etzmuss, and Wolfgang Strasser. "Analysis of numerical methods for the simulation of deformable models". In: *The Visual Computer* 19.7 (Dec. 2003), pp. 581–600. ISSN: 1432-2315. DOI: 10.1007/s00371-003-0206-2.

[69] A. Henriques and B. Wünsche. "Improved Meshless Deformation Techniques for Real-Time Interactive Collaborative Environments". In: *GRAPP '07*. Barcelona, Spain, 2007, pp. 154–161.

[70] Alex Henriques, Burkhard Wünsche, and Stefan Marks. "An investigation of meshless deformation for fast soft tissue simulation in virtual surgery applications". In: (2007).

[71] H.R. Hiester, M.D. Piggott, P.E. Farrell, and P.A. Allison. "Assessment of spurious mixing in adaptive mesh simulations of the two-dimensional lock-exchange". In: *Ocean Modelling* 73.Supplement C (2014), pp. 30 –44. ISSN: 1463-5003. DOI: https://doi.org/10.1016/j.ocemod.2013.10.003.

[72] Ashley Horton, Adam Wittek, and Karol Miller. "Towards Meshless Methods for Surgical Simulation". In: *Computational Biomechanics for Medicine*. Ed. by Karol Miller and Dimos Poulikakos. Copenhagen, Denmark. Vol. 1. MICCAI, 2006, pp. 34–42. ISBN: 10.

[73] Ashley Horton, Adam Wittek, Grand Roman Joldes, and Karol Miller. "A meshless Total Lagrangian explicit dynamics algorithm for surgical simulation". In: *International Journal for Numerical Methods in Biomedical Engineering* 26.8 (2010), pp. 977–998. ISSN: 2040-7947. DOI: 10.1002/cnm.1374.

[74] M Hrapko, JA Van Dommelen, GW Peters, and JS Wismans. "The influence of test conditions on characterization of the mechanical properties of brain tissue". In: *Journal of Biomechanical Engineering* 130.3 (2008), p. 031003.

[75] J.D. Humphrey and F.C. Yin. "A new constitutive formulation for characterizing the mechanical behavior of soft tissues". In: *Biophysical Journal* 52.4 (1987), pp. 563 –570. ISSN: 0006-3495. DOI: http://dx.doi.org/10.1016/S0006-3495(87)83245-9.

[76] Xavier Provot Institut and Xavier Provot. "Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior". In: *In Graphics Interface*. 1996, pp. 147–154.

[77] G. Irving, J. Teran, and R. Fedkiw. "Invertible Finite Elements for Robust Simulation of Large Deformation". In: *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA '04. Grenoble, France: Eurographics Association, 2004, pp. 131–140. ISBN: 3-905673-14-2. DOI: 10.1145/1028523.1028541.

[78] Geoffrey Irving, Joseph Teran, and Ronald Fedkiw. "Invertible finite elements for robust simulation of large deformation". In: *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association. 2004, pp. 131–140.

[79] O. Jarrousse, T. Fritz, and O. Dössel. "Implicit Time Integration in a Volumetric Mass-Spring System for Modeling Myocardial Elastomechanics". In: *World Congress on Medical Physics and Biomedical Engineering, September 7 - 12, 2009, Munich, Germany: Vol. 25/4 Image Processing, Biosignal Processing, Modelling and Simulation, Biomechanics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 876–879. ISBN: 978-3-642-03882-2.

[80] Lenka Jerábková and Torsten Kuhlen. "Stable cutting of deformable objects in virtual environments using XFEM". In: *IEEE Computer Graphics and Applications* 29.2 (2009), pp. 61–71.

[81] L. Jerábková, J. Jerábek, R. Chudoba, and T. Kuhlen. "Stable Interactive Cutting of Deformable Objects". In: *Vriphys: 3rd Workshop in Virtual Realitiy, Interactions, and Physical Simulation*. Ed. by Cesar Mendoza and Isabel Navazo. The Eurographics Association, 2006. ISBN: 3-905673-61-4. DOI: 10.2312/PE/vriphys/vriphys06/125-126.

[82] M. Farshadb J.G. Snedekera, M. Barbezatb, P. Niederera, F.R. Schmidlinc. "Strain energy density as a rupture criterion for the kidney: impact tests on porcine organs, finite element simulation, and a baseline comparison between human and porcine tissues". In: *Journal of Biomechanics* 38 (2005), pp. 993–1001.

[83] Xia Jin, Grand Roman Joldes, Karol Miller, King H. Yang, and Adam Wittek. "Meshless algorithm for soft tissue cutting in surgical simulation". In: *Computer Methods in Biomechanics and Biomedical Engineering* 17.7 (2014). PMID: 22974246, pp. 800–811. DOI: 10.1080/10255842.2012.716829.

[84] A. Joukhadar, F. Garat, and C. Laugier. "Parameter identification for dynamic simulation". In: *Proceedings of International Conference on Robotics and Automation*. Vol. 3. Apr. 1997, 1928–1933 vol.3. DOI: 10.1109/ROBOT.1997.619070.

[85] Peter Kaufmann, Sebastian Martin, Mario Botsch, Eitan Grinspun, and Markus Gross. "Enrichment Textures for Detailed Cutting of Shells". In: *ACM Trans. Graph.* 28.3 (July 2009), 50:1–50:10. ISSN: 0730-0301. DOI: 10.1145/1531326.1531356.

[86] E. Keeve and R. Kikinis. "Deformable modeling of facial tissue". In: *Proceedings of the First Joint BMES/EMBS Conference. 1999 IEEE Engineering in Medicine and Biology 21st Annual Conference and the 1999 Annual Fall Meeting of the Biomedical Engineering Society (Cat. N*. Vol. 1. 1999, p. 502. DOI: 10.1109/IEMBS.1999.802580.

[87] Andrew R. Kemper, Anthony C. Santago, Joel D. Stitzel, Jessica L. Sparks, and Stefan M. Duma. "Biomechanical response of human spleen in tensile loading". In: *Journal of Biomechanics* 45.2 (2012), pp. 348 –355. ISSN: 0021-9290. DOI: http://dx.doi.org/10.1016/j.jbiomech.2011.10.022.

[88] M. Keshavarz and A. Mojra. "Dynamic modeling of breast tissue with application of model reference adaptive system identification technique based on clinical robot-assisted palpation". In: *Journal of the Mechanical Behavior of Biomedical Materials* 51 (2015), pp. 269 –278. ISSN: 1751-6161. DOI: http://dx.doi.org/10.1016/j.jmbbm.2015.07.017.

[89] Junggon Kim and Nancy S Pollard. "Fast simulation of skeleton-driven deformable body characters". In: *ACM Transactions on Graphics (TOG)* 30.5 (2011), p. 121.

[90] Dieter Klatt, Christian Friedrich, Yasmin Korth, Robert Vogt, Jürgen Braun, and Ingolf Sack. "Viscoelastic properties of liver measured by oscillatory rheometry and multifrequency magnetic resonance elastography". In: *Biorheology* 47.2 (2010), pp. 133–141.

[91] M. Kot and H. Nagahashi. "Second Degree of Freedom of Elastic Objects - Adjustable Poisson's Ratio for Mass Spring Models". In: *GRAPP 2015 - Proceedings, Berlin, Germany, 11-14 March, 2015*. SciTePress, 2015, pp. 138–142. ISBN: 978-989-758-087-1. DOI: http://dx.doi.org/10.5220/0005303601380142.

[92] M. Kot, H. Nagahashi, and P. Szymczak. "Elastic moduli of simple mass spring models". In: *The Visual Computer* (Sept. 2014). ISSN: 0178-2789. DOI: 10.1007/s00371-014-1015-5.

[93] Yoram Lanir and Ravi Namani. "Reliability of structure tensors in representing soft tissues structure". In: *Journal of the Mechanical Behavior of Biomedical Materials* 46 (2015), pp. 222 –228. ISSN: 1751-6161. DOI: http://dx.doi.org/10.1016/j.jmbbm.2015.02.012.

[94] Donghua Liao, Lene Hee, Puk Sandager, Niels Uldbjerg, and Hans Gregersen. "Identification of biomechanical properties in vivo in human uterine cervix." In: *Journal of the mechanical behavior of biomedical materials* 39 (Nov. 2014), pp. 27–37. ISSN: 1878-0180. DOI: 10.1016/j.jmbbm.2014.07.005.

[95] P. Lienhardt. "N-Dimensional Generalized Combinatorial Maps and Cellular Quasi-Manifolds". In: *Int. J. Comput. Geometry Appl.* 4.3 (1994), pp. 275–324.

[96] E.M. Lifshitz, A.M. Kosevich, and L.P. Pitaevskii. "Theory of Elasticity (Third Edition)". In: Third Edition. Oxford: Butterworth-Heinemann, 1986. ISBN: 978-0-08-057069-3.

[97] Alex Lindblad and George Turkiyyah. "A Physically-based Framework for Real-time Haptic Cutting and Interaction with 3D Continuum Models". In: *Proceedings of the 2007 ACM Symposium on Solid and Physical Modeling*. SPM '07. New York, NY, USA: ACM, 2007, pp. 421–429. ISBN: 978-1-59593-666-0. DOI: 10.1145/1236246.1236307.

[98] Zizhen Liu and Lynne Bilston. "On the viscoelastic character of liver tissue: experiments and modelling of the linear behaviour". In: *Biorheology* 37.3 (2000), pp. 191–201.

[99] B. A. Lloyd, G. Székely, and M. Harders. "Identification of spring parameters for deformable object simulation". In: *IEEE Transactions on Visualization and Computer Graphics* 13.1 (2007), pp. 1081–1093. ISSN: 10772626. DOI: 10.1109/TVCG.2007.1055.

[100] E. G. Loboa, T. A. L. Wren, G. S. Beaupré, and D. R. Carter. "Mechanobiology of soft skeletal tissue differentiation—a computational approach of a fiber-reinforced poroelastic model based on homogeneous and isotropic simplifications". In: *Biomechanics and Modeling in Mechanobiology* 2.2 (Nov. 2003), pp. 83–96. ISSN: 1617-7940. DOI: 10.1007/s10237-003-0030-7.

[101] Jean Louchet, Xavier Provot, and David Crochemore. "Evolutionary identification of cloth animation models". In: *Computer Animation and Simulation '95: Proceedings of the Eurographics Workshop in Maastricht, The Netherlands, September 2–3, 1995*. Ed. by Demetri Terzopoulos and Daniel Thalmann. Vienna: Springer Vienna, 1995, pp. 44–54. ISBN: 978-3-7091-9435-5. DOI: 10.1007/978-3-7091-9435-5_4.

[102] Anderson Maciel, Ronan Boulic, and Daniel Thalmann. "Deformable Tissue Parameterized by Properties of Real Biological Tissue". In: *Surgery Simulation and Soft Tissue Modeling: International Symposium, IS4TM 2003 Juan-Les-Pins, France, June 12–13, 2003, Proceedings*. Ed. by Nicholas Ayache and Hervé Delingette. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 74–87. ISBN: 978-3-540-45015-3. DOI: 10.1007/3-540-45015-7_8.

[103] Miles Macklin, Matthias Müller, and Nuttapong Chentanez. "XPBD: Position-based Simulation of Compliant Constrained Dynamics". In: *Proceedings of the 9th International Conference on Motion in Games*. MIG '16. San Francisco, USA: ACM, 2016, pp. 49–54. ISBN: 978-1-4503-4592-7. DOI: 10.1145/2994258.2994272.

[104] Miles Macklin, Matthias Müller, Nuttapong Chentanez, and Tae-Yong Kim. "Unified Particle Physics for Real-Time Applications". In: *ACM Transactions on Graphics (TOG)* 33.4 (2014), p. 104.

[105] Maud Marchal. "Soft tissue modeling for computer assisted medical interventions". Theses. Université Joseph-Fourier - Grenoble I, Dec. 2006.

[106] U. Meier, O. López, C. Monserrat, M.C. Juan, and M. Alcañiz. "Real-time deformable models for surgery simulation: a survey". In: *Computer Methods and Programs in Biomedicine* 77.3 (2005), pp. 183 –197. ISSN: 0169-2607. DOI: https://doi.org/10.1016/j.cmpb.2004.11.002.

[107] Sia Meshkat and Dafna Talmor. "Generating a mixed mesh of hexahedra, pentahedra and tetrahedra from an underlying tetrahedral mesh". In: *International Journal for Numerical Methods in Engineering* 49.1-2 (2000), pp. 17–30. ISSN: 1097-0207. DOI: 10.1002/1097-0207(20000910/20)49:1/2<17::AID-NME920>3.0.CO;2-U.

[108] Karol Miller and Kiyoyuki Chinzei. "Constitutive modelling of brain tissue: experiment and theory". In: *Journal of biomechanics* 30.11 (1997), pp. 1115–1121.

[109] Karol Miller, Kiyoyuki Chinzei, Girma Orssengo, and Piotr Bednarz. "Mechanical properties of brain tissue in-vivo: experiment and computer simulation". In: *Journal of biomechanics* 33.11 (2000), pp. 1369–1376.

[110] Karol Miller, Grand Joldes, Dane Lance, and Adam Wittek. "Total Lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation". In: *Communications in Numerical Methods in Engineering* 23.2 (2007), pp. 121–134. ISSN: 1099-0887. DOI: 10.1002/cnm.887.

[111] Sarthak Misra, KT Ramesh, and Allison M Okamura. "Modelling of non-linear elastic tissues for surgical simulation". In: *Computer methods in biomechanics and biomedical engineering* 13.6 (2010), pp. 811–818.

[112] Wouter Mollemans, Filip Schutyser, Johan Van Cleynenbreugel, and Paul Suetens. "Tetrahedral Mass Spring Model for Fast Soft Tissue Deformation". In: *Surgery Simulation and Soft Tissue Modeling: International Symposium, IS4TM 2003 Juan-Les-Pins, France, June 12–13, 2003, Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 145–154. ISBN: 978-3-540-45015-3.

[113] Nicolas Moës, John Dolbow, and Ted Belytschko. "A finite element method for crack growth without remeshing". In: *International Journal for Numerical Methods in Engineering* 46.1 (1999), pp. 131–150. ISSN: 1097-0207. DOI: 10.1002/(SICI)1097-0207(19990910)46:1<131::AID-NME726>3.0.CO;2-J.

[114] Matthias Müller and Markus Gross. "Interactive virtual materials". In: *Proceedings of Graphics Interface 2004*. Canadian Human-Computer Communications Society. 2004, pp. 239–246.

[115] Matthias Müller, Bruno Heidelberger, Matthias Teschner, and Markus Gross. "Meshless Deformations Based on Shape Matching". In: *ACM SIGGRAPH 2005 Papers*. SIGGRAPH '05. Los Angeles, California: ACM, 2005, pp. 471–478. DOI: 10.1145/1186822.1073216.

[116] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. "Position Based Dynamics". In: *J. Vis. Comun. Image Represent.* 18.2 (Apr. 2007), pp. 109–118. ISSN: 1047-3203. DOI: 10.1016/j.jvcir.2007.01.005.

[117] Ahmed Nasim and Jerome Vernick. "Management of liver trauma in adults". In: vol. 4. Jan. 2011, pp. 114–9. DOI: 10.4103/0974-2700.76846.

[118] Simin Nasseri, Lynne E Bilston, and Nhan Phan-Thien. "Viscoelastic properties of pig kidney in shear, experimental results and modelling". In: *Rheologica acta* 41.1 (2002), pp. 180–192.

[119] Suriya Natsupakpong and M. Cenk Çavuşoğlu. "Determination of elasticity parameters in lumped element (mass-spring) models of deformable objects". In: *Graphical Models* 72.6 (2010), pp. 61–73. ISSN: 1524-0703. DOI: http://dx.doi.org/10.1016/j.gmod.2010.10.001.

[120] Alessandro Nava, Edoardo Mazza, M Furrer, Peter Villiger, and WH Reinhart. "In vivo mechanical characterization of human liver". In: *Medical image analysis* 12.2 (2008), pp. 203–216.

[121] A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson. "Physically Based Deformable Models in Computer Graphics". In: *Computer Graphics Forum* 25.4 (Dec. 2006), pp. 809–836. ISSN: 0167-7055.

[122] S. Nicolle, L. Noguer, and J.-F. Palierne. "Shear mechanical properties of the porcine pancreas: Experiment and analytical modelling". In: *Journal of the Mechanical Behavior of Biomedical Materials* 26.Supplement C (2013), pp. 90 –97. ISSN: 1751-6161. DOI: https://doi.org/10.1016/j.jmbbm.2013.05.029.

[123] S. Nicolle, L. Noguer, and J. F. Palierne. "Shear mechanical properties of the spleen: Experiment and analytical modelling". In: *Journal of the Mechanical Behavior of Biomedical Materials* 9 (May 2012), pp. 130–136. ISSN: 1878-0180. DOI: 10.1016/j.jmbbm.2012.02.005.

[124] S Nicolle, P Vezin, and J-F Palierne. "A strain-hardening bi-power law for the nonlinear behaviour of biological soft tissues." In: *Journal of biomechanics* 43.5 (Mar. 2010), pp. 927–32. ISSN: 1873-2380. DOI: 10.1016/j.jbiomech.2009.11.002.

[125] Stéphane Nicolle, Mourad Lounis, and Rémy Willinger. "Shear properties of brain tissue over a frequency range relevant for automotive impact situations: new experimental results". In: *Stapp Car Crash Journal* 48 (2004), p. 239.

[126] Stéphane Nicolle, Johanna Decorps, Bérengère Fromy, and J-F Palierne. "New regime in the mechanical behavior of skin: strain-softening occurring before strain-hardening". In: *Journal of the mechanical behavior of biomedical materials* 69 (2017), pp. 98–106.

[127] Jouko Niiranen. "Fast and accurate symmetric euler algorithm for electromechanical simulations". In: *IMACS* 1 (1999), pp. 71–78.

[128] S. Niroomandi, D. González, I. Alfaro, F. Bordeu, A. Leygue, E. Cueto, and F. Chinesta. "Real-time simulation of biological soft tissues: a PGD approach". In: *International Journal for Numerical Methods in Biomedical Engineering* 29.5 (2013), pp. 586–600. ISSN: 2040-7947. DOI: 10.1002/cnm.2544.

[129] S. Niroomandi, I. Alfaro, D. González, E. Cueto, and F. Chinesta. "Real-time simulation of surgery by reduced-order modeling and X-FEM techniques". In: *International Journal for Numerical Methods in Biomedical Engineering* 28.5 (2012), pp. 574–588. ISSN: 2040-7947. DOI: 10.1002/cnm.1491.

[130] Siamak Niroomandi, Icíar Alfaro, Elías Cueto, and Francisco Chinesta. "Real-time deformable models of non-linear tissues by model reduction techniques". In: *Computer methods and programs in biomedicine* 91.3 (2008), pp. 223–231.

[131] B. Nour-Omid and C.C. Rankin. "Finite rotation analysis and consistent linearization using projectors". In: *Computer Methods in Applied Mechanics and Engineering* 93.3 (1991), pp. 353 –384. ISSN: 0045-7825. DOI: https://doi.org/10.1016/0045-7825(91)90248-5.

[132] Steven J Owen, Scott A Canann, and Sunil Saigal. "Pyramid elements for maintaining tetrahedra to hexahedra conformability". In: *ASME Applied Mechanics Division-Publications-AMD* 220 (1997), pp. 123–130.

[133] Steven J Owen and Sunil Saigal. "Formation of pyramid elements for hexahedra to tetrahedra transitions". In: *Computer Methods in Applied Mechanics and Engineering* 190.34 (2001), pp. 4505 –4518. ISSN: 0045-7825. DOI: http://dx.doi.org/10.1016/S0045-7825(00)00330-3.

[134] Celine Paloc, Fernando Bello, Richard I. Kitney, and Ara Darzi. "Online Multiresolution Volumetric Mass Spring Model for Real Time Soft Tissue Deformation". In: *Medical Image Computing and Computer-Assisted Intervention — MICCAI 2002: 5th International Conference Tokyo, Japan, September 25–28, 2002 Proceedings, Part II*. Ed. by Takeyoshi Dohi and Ron Kikinis. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 219–226. ISBN: 978-3-540-45787-9. DOI: 10.1007/3-540-45787-9_28.

[135] G. Picinbono, H. Delingette, and N. Ayache. "Real-Time Large Displacement Elasticity for Surgery Simulation: Non-linear Tensor-Mass Model". In: *Proceedings of MICCAI'00*. London, UK: Springer-Verlag, 2000, pp. 643–652. ISBN: 978-3-540-40899-4. DOI: 10.1007/978-3-540-40899-4_66.

[136] Alessandro Protti, Emiliano Votta, and Luciano Gattinoni. "Which is the most important strain in the pathogenesis of ventilator-induced lung injury: dynamic or static?" In: *Current Opinion in Critical Care* 20 (1 Feb. 2014), pp. 33–38. DOI: 10.1097/MCC.0000000000000047.

[137] Laurent Rineau and Mariette Yvinec. "3D Surface Mesh Generation". In: *CGAL User and Reference Manual*. 4.10.1. CGAL Editorial Board, 2017. URL: http://doc.cgal.org/4.10.1/Manual/packages.html#PkgSurfaceMesher3Summary.

[138] Jacob Rosen, Jeffrey D Brown, Smita De, Mika Sinanan, and Blake Hannaford. "Biomechanical properties of abdominal organs in vivo and postmortem under compression loads". In: *Journal of biomechanical engineering* 130.2 (2008), p. 021020.

[139] Ichiro Sakuma, Yosuke Nishimura, Chee Chui, Etsuko Kobayashi, Hiroshi Inada, Xian Chen, and Toshiaki Hisada. "In vitro measurement of mechanical properties of liver tissue under compression and elongation using a new test piece holding method with surgical glue". In: *Surgery Simulation and Soft Tissue Modeling* (2003), pp. 1003–1003.

[140] Alessandro Sala, Giuseppe Turini, Mauro Ferrari, Franco Mosca, and Vincenzo Ferrari. "Integration of biomechanical parameters in tetrahedral mass-spring models for virtual surgery simulation". In: *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*. 2011, pp. 4550–4554.

[141] Gaizka San-Vicente, Iker Aguinaga, and Juan Tomas Celigueta. "Cubical Mass-Spring Model Design Based on a Tensile Deformation Test and Nonlinear Material Model". In: *IEEE Transactions on Visualization and Computer Graphics* 18.2 (Feb. 2012), pp. 228–241. ISSN: 1077-2626. DOI: 10.1109/TVCG.2011.32.

[142] J.-M. Schwartz, M. Denninger, D. Rancourt, C. Moisan, and D. Laurendeau. "Modelling liver tissue properties using a non-linear visco-elastic model for surgery simulation." In: *Medical image analysis* 9 (2005), pp. 103–112. ISSN: 13618415. DOI: 10.1016/j.media.2004.11.002.

[143] Kenji Shigeta, Yuichi Kitagawa, and Tsuyoshi Yasuki. "Development of next generation human FE model capable of organ injury prediction". In: *Proceedings of the 21st Annual Enhanced Safety of Vehicles* (2009).

[144] Josildo Pereira da Silva, Gilson A. Giraldi, and Antônio L. Apolinário Jr. "A new optimization approach for mass-spring models parameterization". In: *Graphical Models* 81 (2015), pp. 1 –17. ISSN: 1524-0703. DOI: http://dx.doi.org/10.1016/j.gmod.2015.07.001.

[145] Thomas H. Speeter. "Three-dimensional Finite Element Analysis of Elastic Continua for Tactile Sensing". In: *The International Journal of Robotics Research* 11.1 (1992), pp. 1–19. DOI: 10.1177/027836499201100101.

[146] J. Stingl, V. Báča, P. Čech, J. Kovanda, H. Kovandová, V. Mandys, J. Rejmontová, and B. Sosna. "Morphology and some biomechanical properties of human liver and spleen". In: *Surgical and Radiologic Anatomy* 24.5 (Dec. 2002), pp. 285–289. ISSN: 1279-8517. DOI: 10.1007/s00276-002-0054-1.

[147] J. Su, R. Sheth, and R. Fedkiw. "Energy Conservation for the Simulation of Deformable Bodies". In: *IEEE Transactions on Visualization and Computer Graphics* 19.2 (Feb. 2013), pp. 189–200. ISSN: 1077-2626. DOI: 10.1109/TVCG.2012.132.

[148] N. Sukumar, N. Moës, B. Moran, and T. Belytschko. "Extended finite element method for three-dimensional crack modelling". In: *International Journal for Numerical Methods in Engineering* 48.11 (2000), pp. 1549–1570. ISSN: 1097-0207. DOI: 10.1002/1097-0207(20000820)48:11<1549::AID-NME955>3.0.CO;2-A.

[149] D. Terzopoulos and A. Witkin. "Physically based models with rigid and deformable components". In: *IEEE Computer Graphics and Applications* 8.6 (1988), pp. 41–51.

[150] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. "Elastically deformable models". In: *SIGGRAPH*. 1987, pp. 205–214.

[151] M. Teschner, B. Heidelberger, M. Muller, and M. Gross. "A versatile and robust model for geometrically complex deformable solids". In: *Proceedings Computer Graphics International*. June 2004, pp. 312–319. DOI: 10.1109/CGI.2004.1309227.

[152] Matthias Teschner, Sabine Girod, and Bernd Girod. "Direct Computation of Nonlinear Soft-Tissue Deformation". In: *VMV*. 2000.

[153] C. Truesdell. *A First Course in Rational Continuum Mechanics, Vol 1: General Concepts*. 2nd ed. corr., rev. and augmented. Boston, Mass. : Academic Press, 1991.

[154] JAW Van Dommelen, TPJ Van der Sande, M Hrapko, and GWM Peters. "Mechanical properties of brain tissue by indentation: interregional variation". In: *Journal of the mechanical behavior of biomedical materials* 3.2 (2010), pp. 158–166.

[155] Philippe Vezin and Jean Pierre Verriest. "Development of a set of numerical human models for safety". In: *The 19th International ESV Conference*. 2005, pp. 05–0163.

[156] G. San Vincente-Otamendi. "Designing deformable models of soft tissue for virtual surgery planning and simulation using the Mass-Spring Model". PhD thesis. Universidad de Navarra, 2011.

[157] W. Wang, X. Yan, Y. Xie, J. Qin, W. M. Pang, and P. A. Heng. "A Physically-Based Modeling and Simulation Framework for Facial Animation". In: *2009 Fifth International Conference on Image and Graphics*. Sept. 2009, pp. 521–526. DOI: 10.1109/ICIG.2009.26.

[158] Carley Conrad Ward and Robert B. Thompson. "The Development of a Detailed Finite Element Brain Model". In: *AE Technical Paper*. SAE International, Feb. 1975. DOI: 10.4271/751163.

[159] Gerald Wempner. "Finite elements, finite rotations and small strains of flexible shells". In: *International Journal of Solids and Structures* 5.2 (1969), pp. 117 –153. ISSN: 0020-7683. DOI: https://doi.org/10.1016/0020-7683(69)90025-0.

[160] Jane Wilhelms and Allen Van Gelder. "Anatomically Based Modeling". In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '97. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1997, pp. 173–180. ISBN: 0-89791-896-7. DOI: 10.1145/258734.258833.

[161] Han-Chin Wu and Ren-Feng Yao. "Mechanical behavior of the human annulus fibrosus". In: *Journal of Biomechanics* 9.1 (1976), pp. 1 –7. ISSN: 0021-9290. DOI: http://dx.doi.org/10.1016/0021-9290(76)90132-9.

[162] Xiaofeng Wu. "Reduced Deformable Body Simulation with Richer Dynamics". PhD thesis. The Ohio State University, 2016.

[163] Xunlei Wu, Michael S. Downes, Tolga Goktekin, and Frank Tendick. "Adaptive Nonlinear Finite Elements for Deformable Body Simulation Using Dynamic Progressive Meshes". In: *Computer Graphics Forum* 20.3 (2001), pp. 349–358. ISSN: 1467-8659. DOI: 10.1111/1467-8659.00527.

[164] Soji Yamakawa and Kenji Shimada. "Converting a tetrahedral mesh to a prism–tetrahedral hybrid mesh for FEM accuracy and efficiency". In: *International Journal for Numerical Methods in Engineering* 80.1 (2009), pp. 74–102. ISSN: 1097-0207. DOI: 10.1002/nme.2634.

[165] Soji Yamakawa and Kenji Shimada. "Fully-automated hex-dominant mesh generation with directionality control via packing rectangular solid cells". In: *International Journal for Numerical Methods in Engineering* 57.15 (2003), pp. 2099–2129. ISSN: 1097-0207. DOI: 10.1002/nme.754.

[166] Narayan Yoganandan, Frank Pintar, Thomas Gennarelli, and Matthew Maltese. "Patterns of Abdominal Injuries in Frontal and Side Impacts". In: *Annual Proceedings / Association for the Advancement of Automotive Medicine*. Vol. 44. 2000, pp. 17–36.

[167] D. Zerbato, S. Galvan, and P. Fiorini. "Calibration of mass spring models for organ simulations". In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Oct. 2007, pp. 370–375. DOI: 10.1109/IROS.2007.4399289.

[168] Jun Zhang, Guan-chang Jin, Li-Bo Meng, Longhui Jian, Aiyuan Wang, and Shibi Lu. "Strain and mechanical behavior measurements of soft tissues with digital speckle method". In: *Journal of Biomedical Optics* 10 (2005), pp. 10 –10 –7. DOI: 10.1117/1.1895185.

[169] Gao-Feng Zhao. "Developing a four-dimensional lattice spring model for mechanical responses of solids". In: *Computer Methods in Applied Mechanics and Engineering* 315 (2017), pp. 881–895.

[170] Gao-Feng Zhao, Jiannong Fang, and Jian Zhao. "A 3D distinct lattice spring model for elasticity and dynamic failure". In: *International Journal for Numerical and Analytical Methods in Geomechanics* 35.8 (2011), pp. 859–885. ISSN: 1096-9853. DOI: 10.1002/nag.930.

[171] Shuan-Feng Zhao and Gao-Feng Zhao. "Implementation of a high order lattice spring model for elasticity". In: *International Journal of Solids and Structures* 49.18 (2012), pp. 2568–2581.

[172] Boran Zhou, Alexander Rachev, and Tarek Shazly. "The biaxial active mechanical properties of the porcine primary renal artery". In: *Journal of the Mechanical Behavior of Biomedical Materials* 48 (2015), pp. 28 –37. ISSN: 1751-6161. DOI: http://dx.doi.org/10.1016/j.jmbbm.2015.04.004.