

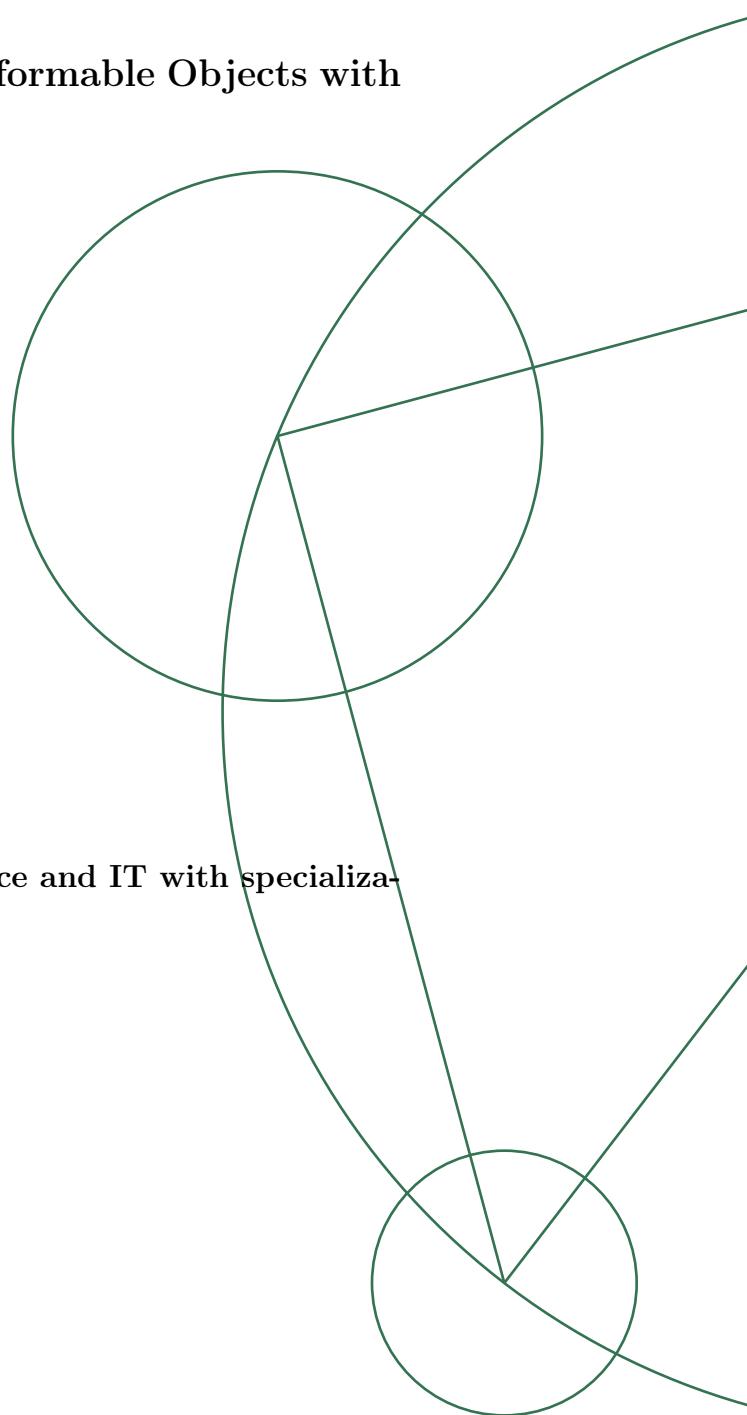
August 11, 2016



Implicit Method for Simulation of Deformable Objects with a Large Stiffness

A study on efficiency by numerical approximation

Amalie Therese Willum Hansen



Bachelor project for B.Sc. in Natural Science and IT with specialization in Mathematics

Advisor: Kenny Erleben

Abstract

The main focus in this B.Sc. project is a physically based model for simulation deformable bodies with a large stiffness. A numerical approximation for the stiffness tensor, will be applied instead of the exact version. It is known from previous work [9] that such an approximation is applicable to a non-physically based deformable model.

There will be an experimental validation of whether or not the numerical approximation of the stiffness term is also applicable to a real physical deformable model. The experiments consist of crossvalidations of the essential parameters, especially the stiffness measure, Young's Modulus, and the time step size is in focus. Crossvalidation of the parameters will supply us a feasible area, constructed from a user-decided tolerance. The findings from the crossvalidation showed a feasible area for Young's Modulus and time step size. For a tolerance equal to 10^{-3} , we found that for an unrealistic soft material, the needed time step size is an order in magnitude of 10^{-3} and for a highly stiff material, the time step size should be at least 10^{-9} . These results indicate that the numerical approximation for the stiffness tensor, is not applicable to a physical model, when envolving a real value for Young's Modulus.

Resumé

Det primære fokus i dette bachelorprojekt er en fysisk baseret model med det formål at kunne simulere stive deformerbare objekter. En numerisk approksimation af stivhedstensoren vil blive brugt i stedet for den eksakte version deraf. Der vides fra tidligere arbejde [9] at en numerisk approksimation kan anvendes, når man bruger en ikke-fysisk baseret deformbar model. Der vil være eksperimetiel validering om hvorvidt man kan anvende den numeriske approksimation i ægte fysisk baseret model.

Eksperimenterne vil bestå af krydsvalideringer af de mest essentielle parametre, specielt vil stivheds parameteren, Young's Modulus og tidsskridstørrelsen være i fokus. Krydsvalideringen vil forsyne os med et område, hvori det ville være muligt at vælge værdier til sine parametre, udfra en bruger bestemt tolerance. Resultaterne fra krydsvalideringen vidste et område for Young's Modulus og tidsskridstørrelsen. Med en tolerance på 10^{-3} , fandt vi at for en urealistisk blødt materiale skal man minimum have en tidsskridtstørrelse af en orden på 10^{-3} og for et meget stift materiale skal man minimum have en tidsskridtstørrelse af ordenen 10^{-9} . Disse resultater kunne indikere at en numerisk approksimation af stivhedstensoren ikke kan anvendes i en fysisk baseret model, hvor der bruges ægte værdier for Young's Modulus.

Contents

1	Introduction	1
2	Tensor Algebra	1
2.1	Zeroth and First Order Tensors	2
2.2	Second Order Tensors	2
2.3	Properties of Tensors	3
2.3.1	The Identity Tensor	4
2.3.2	The Trace of a Tensor	4
2.3.3	The Transpose of a Tensor	4
2.3.4	The Inverse of a Tensor	4
2.3.5	Summation, Multiplication and Double Contraction of Tensors .	4
2.3.6	The ∇ Operator on a Tensor	5
3	Continuum Mechanics	5
3.1	The Displacement Field	6
3.2	The Deformation Gradient	6
3.2.1	Simple Examples of the Deformation Map Configurations	7
3.3	Stress Tensors	8
3.4	Constitutive Models	9
3.4.1	Strain Tensors	9
3.4.2	Linear Elastic Material	10
3.4.3	Saint Venant-Kirchhoff Material	11
4	The Finite Element Method	13
4.1	Weak Formulation	13
4.2	Shape Functions	15
4.3	Nonlinear Finite Element Method	16
5	Discretization	17
5.1	Implicit Time Integration by Backward Euler	17
5.2	Approximation of the Stiffness Tensor	20
6	Implementation	21
6.1	A Matrix-Free Solution	21
6.2	Computing the Stiffness Tensor	21
7	Experiments	23
7.1	Properties for Using an Iterative Solver	24
7.2	Order Of Approximation	25
7.3	Crossvalidation of Parameters	27
7.4	2D Experiments	33
8	Conclusion	37
9	Further Work	38
	References	39

Appendices	40
A Symbolic Differentiation	40
B Crossvalidation: $\nu \times h$	41
C Crossvalidation: $L \times h$	42

List of Figures

1 Illustrative Examples of Deformation Maps	7
2 St. Venant-Kirchhoff Material Model Cubic Polynomial	12
3 Deformation of a Linear Tetrahedron	16
4 Simulation of Deformation with Approximation and Exact value of the Stiffness Tensor after 1.5 seconds	23
5 The Elastic Forces as a Function of τ	26
6 Crossvalidation of E and h with unit test vectors and random test vectors on 1st order approximation	28
7 Crossvalidation of E and h with “real” test vectors on 1st and 2nd order approximation	30
8 Crossvalidation of ν and h with “real” test vectors on 1st and 2nd order approximation	31
9 Crossvalidation of L and h with “real” test vectors on 1st order approximation of the stiffness tensor	32
10 2D view of Crossvalidation of E and h , for fixed value of h	34
11 2D view of Crossvalidation of ν and h , for fixed value of h	35
12 2D view of Crossvalidation of L and h , for fixed value of h	36
13 2D view of Crossvalidation of L and h , for fixed value of L	37
14 Crossvalidation of ν and h with unit test vectors and random test vectors on 1st order approximation	41
15 Crossvalidation of L and h with unit test vectors and random test vectors on 1st order approximation	42

List of Tables

1 Symbolic Differentiation of The First Piola Stress Tensor in MATLAB	22
2 Symbolic Computation of The Stiffness Tensor in MATLAB	22
3 Symmetry of $\mathbf{A}(\bar{v})$ by the Jacobian \mathbf{J}	24
4 Mesh No. Properties	33
5 Table of Correspondance Between Colors, Models and Test Vectors	33
6 Symbolic Differentiation of The Material Elasticity Tensor in MATLAB	40

Preface

This project of 15 ECTS credits is written in a span of six months as a part time project, from the 15th of February to the 12th of August 2016. It is a final project of the B.Sc. in Natural Science and IT with specialization in Mathematics. The project is written under the supervision of Kenny Erleben, of which I owe a great many thanks for always being patient with me and my endless questioning.

Due to a set of unforeseen obstacles, it was not an option to finalize a full project on time. The obstacles include a change in main focus of the project. Thus, going from comparison of numerical approximation and exact version to finding out why the numerical approximation does not work as expected.

1 Introduction

Deformable models is a highly used tool in the field of computational science. These models are used to simulate anything from Disney movies to building constructions. The deformable model can be physically based or non-physically based. The non-physical model has a wider acceptance within offline and interactive animation, as it is easier to implement and computationally cheaper – one would not have to use a real-valued stiffness measures , which is what makes a real physical model more expensive, but it is also what makes it more correct. [2]

The focus during this project is a physically based modeling of continuum bodies, with a large stiffness. Though, this is not impossible with the means of what we already know, but wouldn't it be cool, if we were able to do it more efficiently by approximating the stiffness tensor?

Initially, we would have liked to compare performance through convergence rates and wall clock time etc. between simulations using the exact version of the stiffness tensor, and a numerical approximation of the stiffness tensor, but when the approximation did not award us with any results, the main focus changed into finding out why.

We know from previous work in [9] that a numerical approximation of the stiffness tensor can be applied instead of repeatedly re-computing the stiffness term. If we were able to do this, we could obtain the benefits from a physical model, and maybe only by the means from a non-physical model. Is it possible to have the best of both worlds? Can we even support the results from [9], by finding a feasible area for the stiffness term?

There will be laid out the mathematical preliminaries, consisting of the most important properties of tensor algebra in section 2. This knowledge will lead to an understanding of the mathematical and theoretical arguments stating the physical assumptions in section 3, which later used to perform a rigorous derivation of the discretization method, implicit forward Euler in section 5.

In order to find out whether or not the finite difference approximation is advantageous, several experiments is performed in section 7. These experiments consist of crossvalidation with the aim of estimating feasible areas for specific tolerances and parameter selections. All experimental implementation is done in MATLAB with help from the framework Hyper-sim [3] provided by my advisor.

2 Tensor Algebra

In the paper we are going to use tensors as a mathematical tool with the purpose of describing the theory behind continuum mechanics. The most basic concepts of vectors and scalars should already be known, though maybe not as in the format of tensors. Hence a concise introduction of notation and terminology of tensor algebra will be carried out.

2.1 Zeroth and First Order Tensors

In tensor notation there is no distinction between row and column vectors opposed to matrix notation, and thus it is coordinate independent [1]. Therefore tensor notation benefits from the fact that it is independent of which kind of coordinate format the user chooses to work with and thus simpler to higher dimensions. The order of the tensor is corresponding to how we distinguish between scalars, vectors and matrices. A scalar is equivalent to a zeroth order tensor and a vector is equivalent to a first order tensor.

By vectors we mean a quantity with specified magnitude and direction in a three dimensional space. A vector is denoted \bar{v} . A vector in tensor notation is defined as

$$\bar{v} = \sum_{i=1}^3 v_i \bar{e}_i \quad (2.1)$$

where v_i is the i 'th component in the vector \bar{v} and \bar{e}_i denotes the cartesian basis vectors. The dot product of basis vectors \bar{e}_i is by definition

$$\delta_{ij} = \bar{e}_i \cdot \bar{e}_j = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases} \quad (2.2)$$

where δ_{ij} is the Kronecker delta function. Therefore we introduce the dot product between two vectors. If \bar{u} and \bar{v} are vectors their dot product will as a result of defintion (2.1) and (2.2) be as follows

$$\bar{u} \cdot \bar{v} = \left(\sum_{i=1}^3 u_i \bar{e}_i \right) \cdot \left(\sum_{j=1}^3 v_j \bar{e}_j \right) \quad (2.3a)$$

$$= \sum_{i=1}^3 \sum_{j=1}^3 u_i v_j \bar{e}_i \cdot \bar{e}_j \quad (2.3b)$$

$$= \sum_{i=1}^3 \sum_{j=1}^3 u_i v_j \delta_{ij} \quad (2.3c)$$

Where the use of the Kronecker delta function will only leave the diagonal of $\bar{u}\bar{v}^T$ left, which is equivalent to

$$= \sum_{i=1}^3 u_i v_i \quad (2.3d)$$

There are physical quantities that are not well represented by vectors in a three dimensional space, some of which can be represented by tensors [1]. Thus we introduce the second order tensor, which we will focus on during the paper.

2.2 Second Order Tensors

A second order tensor in three dimensions is equivalent with a 3×3 square matrix.

From now on vectors will be referred to exclusively as first order tensors.

A second order tensor is a linear transformation between first order tensors, and is defined as

$$\bar{v} = \mathbf{A}\bar{u} \quad (2.4)$$

where \bar{v} and \bar{u} is first order tensors and \mathbf{A} is a second order tensor which is a linear combination of dyadic products

$$\mathbf{A} = \sum_{i=1}^3 \sum_{j=1}^3 A_{ij}(\bar{e}_i \otimes \bar{e}_j) \quad (2.5)$$

For some first order tensors \bar{u} , \bar{w} and \bar{v} the dyadic product is defined as

$$(\bar{u} \otimes \bar{v})\bar{w} = (\bar{v} \cdot \bar{w})\bar{u} \quad (2.6)$$

where $\bar{u} \otimes \bar{v} = \bar{u}\bar{v}^T$.

From the two definitions of first and second order tensors (2.1) and (2.5) it is desirable to substitute into equation (2.4), thus giving us

$$\bar{v} = \mathbf{A}\bar{u} \quad (2.7a)$$

$$= \left(\sum_{i=1}^3 \sum_{j=1}^3 A_{ij}(\bar{e}_i \otimes \bar{e}_j) \right) \left(\sum_{k=1}^3 u_k \bar{e}_k \right) \quad (2.7b)$$

$$= \sum_{i=1}^3 \sum_{j=1}^3 \sum_{k=1}^3 A_{ij} u_k (\bar{e}_i \otimes \bar{e}_j) \bar{e}_k \quad (2.7c)$$

$$= \sum_{i=1}^3 \sum_{j=1}^3 \sum_{k=1}^3 A_{ij} u_k (\bar{e}_j \cdot \bar{e}_k) \bar{e}_i \quad (2.7d)$$

$$= \sum_{i=1}^3 \sum_{j=1}^3 A_{ij} u_j \bar{e}_i. \quad (2.7e)$$

which is the formula for a linear transformation between two first order tensors and a second order tensor.

2.3 Properties of Tensors

Some of the properties of second order tensors are especially the identity tensor, the trace, the transpose and the inverse of a tensor. Summation, multiplication and double contraction is also applicable for tensors, This also applies to the gradient and the divergence of a tensor, given by the operator ∇ .

2.3.1 The Identity Tensor

The linear transformation that transforms every first order tensor into itself is called the identity tensor. The identity tensor is denoted \mathbf{I} , and is defined as

$$\mathbf{I} = \sum_{i=1}^3 (\bar{e}_i \otimes \bar{e}_i) = \sum_{i=1}^3 \sum_{j=1}^3 \delta_{ij} \bar{e}_i \otimes \bar{e}_j \quad (2.8)$$

thus having the property of mapping a vector to itself

$$\bar{v} = \mathbf{I}\bar{v}. \quad (2.9)$$

2.3.2 The Trace of a Tensor

The trace of tensor \mathbf{A} is given by

$$\text{tr}(\mathbf{A}) = \sum_{i=1}^3 A_{ii} \quad (2.10)$$

The trace of a second order tensor \mathbf{A} is defined as the sum of the diagonal components of \mathbf{A} .

2.3.3 The Transpose of a Tensor

Transpose of a second order tensor \mathbf{A}

$$\bar{u} \cdot \mathbf{A}\bar{v} = \bar{v} \cdot \mathbf{A}^T \bar{u} \quad (2.11)$$

In particular if \mathbf{A} is symmetric we write $\mathbf{A} = \mathbf{A}^T$.

2.3.4 The Inverse of a Tensor

Inverse of a second order tensor \mathbf{A} exists if and only if $\det(\mathbf{A}) \neq 0$, and we write

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I} \quad (2.12)$$

Especially if $\mathbf{A}^{-1} = \mathbf{A}^T$, then \mathbf{A} is orthogonal and we write

$$\mathbf{A}^T\mathbf{A} = \mathbf{A}\mathbf{A}^T = \mathbf{I} \quad (2.13)$$

2.3.5 Summation, Multiplication and Double Contraction of Tensors

Summation and multiplication of two second order tensors \mathbf{A} and \mathbf{B} is defined as the two following

$$(\mathbf{A} + \mathbf{B})\bar{v} = \mathbf{A}\bar{v} + \mathbf{B}\bar{v}. \quad (2.14)$$

$$\mathbf{AB}\bar{v} = \mathbf{A}(\mathbf{B}\bar{v}). \quad (2.15)$$

In equation (2.14) it is seen that the distributive laws are applicable to tensors and in equation (2.15) it is seen that tensors are also multiplicative commutative.

Double contraction of two tensors \mathbf{A} and \mathbf{B} is defined as

$$\mathbf{A} : \mathbf{B} = \sum_{i=1}^3 \sum_{j=1}^3 A_{ij} B_{ij} \Leftrightarrow \text{tr}(\mathbf{A}^T \mathbf{B}) = \text{tr}(\mathbf{B}^T \mathbf{A}) = \mathbf{A} : \mathbf{B} \quad (2.16)$$

2.3.6 The ∇ Operator on a Tensor

The gradient is denoted by the use of the operator ∇ . The gradient of a first order tensor is then defined as

$$(\nabla \bar{b})_{ij} = \partial_j b_i \quad (2.17)$$

The result of the gradient of first order tensor will be a second order tensor.

The divergence operator mapped onto a first order tensor is denoted $\nabla \cdot \bar{b}$ and is defined as

$$\nabla \cdot \bar{b} = \sum_i \partial_i b_i \Leftrightarrow \text{tr}(\nabla \bar{b}) = \nabla \cdot \bar{b} \quad (2.18)$$

The result of the divergence of a first order tensor is mapped into a zeroth order tensor.

The divergence operator is also defined for second order tensors as

$$\nabla \cdot \mathbf{A} = \sum_i \sum_j \partial_j A_{ji} \bar{e}_i \quad (2.19)$$

During this section there have been presented some of the fundamentals within tensor algebra.

The algebraic preliminaries presented in this section are going to be used in order to give an explanatory and intuitive understanding of the central concepts of continuum mechanics during the next section.

3 Continuum Mechanics

Continuum mechanics is a specific section of mechanics. Behind the word, continuum, lies an acceptance of infinitesimal volume of materials, meaning that it assumes a body to be infinitely divisible, why it is applicable to various computational physical phenomena.

The fundamental principles that continuum mechanics is build around is mainly the general principles of mechanics and axioms thereof, such as conservation of mass, balance of linear momentum and constitutive equations defining idealized materials. [7]

In this section the reader is going to be provided with the intuition behind continuum mechanics that are relevant for hyper-elastic simulation. This will include displacement field, deformation gradient, stress and strain tensors, their relation given by a derivation of the constitutive models and their governing equations for both linear and nonlinear elastic materials. The knowledge obtained for this section is primarily from my advisers unpublished book [4], Bonet and Wood [1], Sifakis course notes [11], but also [5], [7].

3.1 The Displacement Field

The displacement is describing the distance from the material coordinate to the spatial coordinate of one node. When we refer to the displacement field we are simply meaning that it applies to all nodes instead of just one node.

The displacement field is defined as following

$$\bar{u} = \bar{x} - \bar{X} \quad (3.1)$$

where \bar{x} is the spatial coordinates and \bar{X} is the material coordinates.

The cause of displacement for a node is the change in deformation. Hence an introduction to the deformation gradient \mathbf{F} .

3.2 The Deformation Gradient

The deformation gradient is a mapping $\Phi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, meaning a mapping in a three dimensional space from material coordinates to spatial coordinates. Hence the deformation map is

$$\bar{x} = \Phi(\bar{X}) \quad (3.2)$$

The deformation gradient is the derivative of each component of the spatial coordinates, \bar{x} , with respect to the reference frame, \bar{X} , thus it is the Jacobian of the deformation map. The deformation gradient is defined as following

$$\mathbf{F} = \frac{\partial \Phi(\bar{X})}{\partial \bar{X}} = \frac{\partial \bar{x}}{\partial \bar{X}} \quad (3.3)$$

which due to the displacement field, \bar{u} , in equation (3.1) we can write as

$$\bar{x} = \bar{u} + \bar{X} \quad (3.4)$$

thus leading to following, by substituting (3.4) into (3.3)

$$\mathbf{F} = \frac{\partial (\bar{u} + \bar{X})}{\partial \bar{X}} = \frac{\partial \bar{u}}{\partial \bar{X}} + \frac{\partial \bar{X}}{\partial \bar{X}} = \frac{\partial \bar{u}}{\partial \bar{X}} + \mathbf{I} \quad (3.5)$$

3.2.1 Simple Examples of the Deformation Map Configurations

Figure 1: Four examples of deformations in two dimensions

In this section some simple examples are presented with the aim of getting a better understanding of the underlying concept of deformation in continuum mechanics. The figures are drawn in L^AT_EX using TikZ [16] and with inspiration from Sifakis Course notes [11].

In figure 1 there are depicted some simple examples in two dimensions of how the deformation map would look like in the given scenarios of rotation, translation, anisotropic and uniform scaling. In these simple cases it is possible to write $\bar{x} = \Phi(\bar{X})$ in a closed form. The first figure 1a is merely the reference frame consisting of all the material coordinates – all the initial values. The second figure 1b is a configuration of a rotation done by a rotation matrix, with some angle θ , and the point is the material point in 1a, that has been deformed into a spatial point. The deformation gradient and map will be

$$\begin{bmatrix} x \\ y \end{bmatrix} = \Phi(\bar{X}) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} \implies \mathbf{F} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (3.6)$$

for some angle θ .

Figure 1c is depicting a configuration of translated coordinates. This is the simplest deformation scenario, because this is a result of adding a constant vector, $\bar{\alpha}$, to the material coordinates, hence the deformation map and gradient will be

$$\begin{bmatrix} x \\ y \end{bmatrix} = \Phi(\bar{X}) = \begin{bmatrix} X + \alpha_1 \\ Y + \alpha_2 \end{bmatrix} \implies \mathbf{F} = \frac{\partial \Phi(\bar{X})}{\partial \bar{X}} = \mathbf{I} \quad (3.7)$$

I have used the values $\alpha_1 = 5$ and $\alpha_2 = 6$.

Figure 1d is an anisotropic scaling meaning that the reference configurations are scaled differently on the vertical axis than the horizontal axis, this is achieved by multiplying with a vector, $\bar{\alpha}$ with $\alpha_1 \neq \alpha_2$. This example gives the following deformation map and gradient

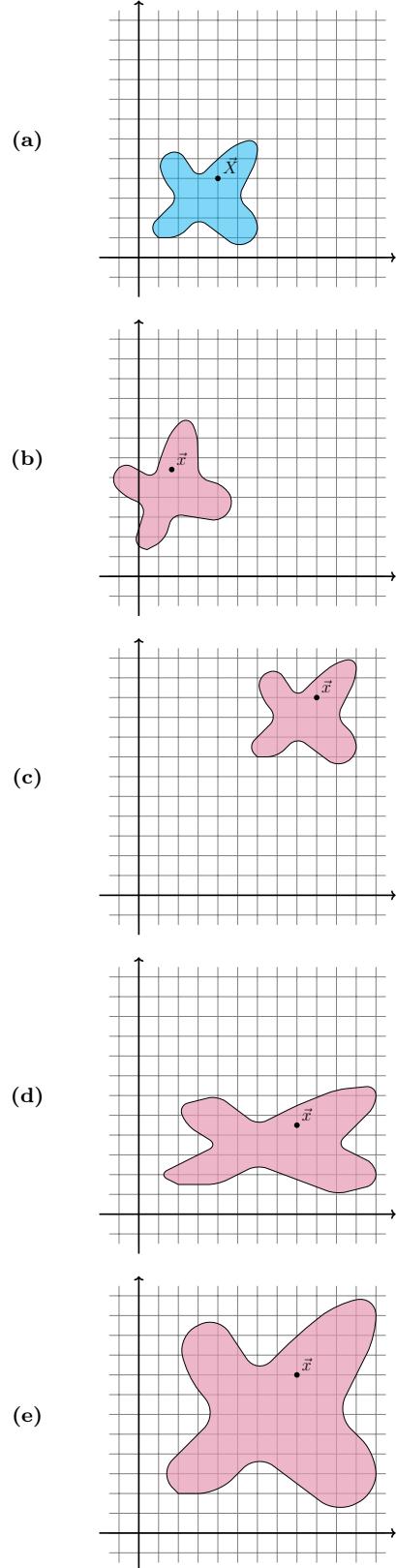
$$\begin{bmatrix} x \\ y \end{bmatrix} = \Phi(\bar{X}) = \begin{bmatrix} X\alpha_1 \\ Y\alpha_2 \end{bmatrix} \implies \mathbf{F} = \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix} \quad (3.8)$$

I have used the values $\alpha_1 = 2$ and $\alpha_2 = 0.5$.

The last figure 1e is very similar to the previous – that being anisotropic scaling – this is uniform scaling. Instead of scaling the horizontal and vertical axes differently, they are scaled uniformly – hence the name – by a single constant γ . Therefore the deformation map and gradient will be

$$\begin{bmatrix} x \\ y \end{bmatrix} = \Phi(\bar{X}) = \begin{bmatrix} X\gamma \\ Y\gamma \end{bmatrix} \implies \mathbf{F} = \gamma \mathbf{I} \quad (3.9)$$

I have used $\gamma = 2$.



3.3 Stress Tensors

The stress tensor can be interpreted as surface tractions that act on a defined reference plane. These are typically decomposed into three orthogonal components. One component is normal to the surface and represents direct stress, and the other two components are tangential to the surface and represent the shear stresses.

The direct stress tends to change the volume of the material and shear stresses tend to deform the material without changing its volume.

Given a spatial surface element

$$d\bar{s} = ds_i \quad (3.10)$$

which means the projection of the area of the surface $d\bar{s}$ onto the i 'th component. We can now form Cauchy's stress hypothesis, that says "*stress is force per unit area*" [4] by applying the force $d\bar{f}$ to the surface element $d\bar{s}$, which leads to

$$d\bar{f}_i = \sum_{j=1}^3 \sigma_{ij} ds_j, \quad \text{for } i \in \{1, 2, 3\} \quad (3.11)$$

where σ is Cauchy's stress tensor and is depending on time and position. This becomes

$$d\bar{f} = \sigma d\bar{s}, \quad (3.12)$$

The first Piola-Kirchoff stress tensor, \mathbf{P} , is defined

$$d\bar{f} = \mathbf{P} d\bar{S} \quad (3.13)$$

thus we can write as following by Cauchy's stress tensor, (3.12)

$$\sigma d\bar{s} = d\bar{f} = \mathbf{P} d\bar{S} \quad (3.14)$$

And by Nanson's Relation¹

$$\sigma d\bar{s} = \sigma j \mathbf{F}^{-1} d\bar{S} = d\bar{f} = \mathbf{P} d\bar{S} \quad (3.15)$$

where $j = \det(\mathbf{F})$. From this we obtain the relation between Cauchy's stress tensor and the first Piola-Kirchhoff stress tensor

$$\mathbf{P} = j \sigma \mathbf{F}^{-1} \quad (3.16)$$

The second Piola-Kirchhoff stress tensor, \mathbf{S} is defined as

$$d\bar{F} = \mathbf{S} d\bar{S} \quad (3.17)$$

and is derived by the same procedure as the first Piola-Kirchhoff stress tensor, thus the result of the second Piola-Kirchoff stress tensor is

$$\mathbf{S} = j \mathbf{F}^{-1} \sigma \mathbf{F}^{-1} \quad (3.18)$$

If we divide Cauchy's stress tensor with the area surface $\|d\bar{s}\|$, the result will be in terms of the surface traction and the surface normal

$$\bar{t} = \sigma \bar{n} \quad (3.19)$$

where the surface traction $\bar{t} = \frac{d\bar{f}}{\|d\bar{s}\|}$ and the surface normal $\bar{n} = \frac{d\bar{s}}{\|d\bar{s}\|}$.

¹The Nanson's Relation is used in expressing area changes from material to spatial coordinates. The formula used: $da \bar{n} = j dA \mathbf{F}^{-1} \bar{N}$. https://en.wikipedia.org/wiki/Finite_strain_theory

3.4 Constitutive Models

Every constitutive model of material is distinctive from one another, by how the physical properties are outlined in the governing equations. Which means that a constitutive model for a specified material gives a relation between the stress and strain measurements [1]

In this section there will be presented a two different constitutive models for linear and nonlinear materials, and their governing equations to relate the deformation to the response the deformation provoke – the relation between stress and strain.

3.4.1 Strain Tensors

The change in deformation is a quantity of how far the current configuration of spatial points are from the reference configuration of material points. This measurement is represented by different strain tensors, depending on what properties one wishes to obtain.

If we express the deformation change in terms of material coordinates, we obtain the Green strain tensor \mathbf{E} defined as

$$\mathbf{E} = \frac{1}{2} (\mathbf{F}^T \mathbf{F} - \mathbf{I}) \quad (3.20)$$

This is a tensor of valuable qualities, i.e. when the elastic body is in its reference frame, the Green strain tensor \mathbf{E} would be equal to $\mathbf{0}$, meaning that no strain occurs, this given by the deformation map $\bar{\Phi}(\bar{X}) = \bar{X}$, this leads to the deformation gradient $\mathbf{F} = \frac{\partial \bar{X}}{\partial X} = \mathbf{I}$ and hence the Green strain tensor $\mathbf{E} = \frac{1}{2} (\mathbf{I}^T \mathbf{I} - \mathbf{I}) = \mathbf{0}$. The same thing is true for when the body exclusively rotates due to $\bar{\Phi}(\bar{X}) = \mathbf{R}\bar{X} + \bar{t} \Rightarrow \mathbf{F} = \frac{\partial(\mathbf{R}\bar{X} + \bar{t})}{\partial X} = \mathbf{R} \Rightarrow \mathbf{E} = \frac{1}{2} (\mathbf{R}^T \mathbf{R} - \mathbf{I}) = \mathbf{0}$. This also means that it is rotation invariant. Admittedly, this is a very useful property but it is also what gives a downside to the Green strain tensor as this comes with the price of a quadratic term, when working with large deformations. This will naturally inflict the constitutive models that make use of the Green strain tensor.

We obtain an explicit expression for the Green strain tensor by substituting equation (3.5) into equation (3.20)

$$\mathbf{E} = \left(\left(\frac{\partial \bar{u}}{\partial \bar{X}} + \mathbf{I} \right)^T \left(\frac{\partial \bar{u}}{\partial \bar{X}} + \mathbf{I} \right) - \mathbf{I} \right) \quad (3.21)$$

$$= \frac{1}{2} \left(\frac{\partial \bar{u}}{\partial \bar{X}}^T \frac{\partial \bar{u}}{\partial \bar{X}} + \frac{\partial \bar{u}}{\partial \bar{X}} + \frac{\partial \bar{u}}{\partial \bar{X}}^T \right). \quad (3.22)$$

By the assumption of infinitesimal strain theory, or small displacement gradient theory, [14] we would have the Cauchy strain tensor $\boldsymbol{\varepsilon}$

$$\boldsymbol{\varepsilon} = \frac{1}{2} (\mathbf{F} + \mathbf{F}^T) - \mathbf{I} \quad (3.23)$$

which also goes by the name of infinitesimal strain tensor or small strain tensor.

As mentioned, we achieve this strain tensor by the assumption of small displacements, which in other words corresponds to a linearization of the Green strain tensor, because we neglect the nonlinear terms

$$\left\| \frac{\partial \bar{u}}{\partial \bar{X}} \right\| \ll 1 \wedge \left\| \frac{\partial \bar{u}^T}{\partial \bar{X}} \right\| \ll 1 \Rightarrow \frac{\partial \bar{u}^T}{\partial \bar{X}} \frac{\partial \bar{u}}{\partial \bar{X}} \approx 0 \quad (3.24)$$

and

$$\bar{x} \approx \bar{X} \Rightarrow \frac{\partial \bar{u}}{\partial \bar{x}} \approx \frac{\partial \bar{u}}{\partial \bar{X}} \Rightarrow \mathbf{E} \approx \boldsymbol{\varepsilon} \quad (3.25)$$

Thus the explicit version of the Cauchy strain tensor will be written out as

$$\boldsymbol{\varepsilon} = \frac{1}{2} \left(\frac{\partial \bar{u}}{\partial \bar{X}} + \frac{\partial \bar{u}^T}{\partial \bar{X}} \right) \quad (3.26)$$

So by approximating the Green strain tensor, we obtain the Cauchy strain tensor, and by doing so we gain a much less complex strain than the Green strain tensor. The reason why it is not near as complex as the Green strain tensor, is in fact that it is limited to only handle infinitesimal rotations. The computed strains will lose accuracy if the rotations become finite. I.e. if the body had only rotated $\bar{\Phi}(\bar{X}) = \mathbf{R}\bar{X} + \bar{t}$ the Cauchy strain would produce a non-zero strain, even though the body had not deformed in shape.

Given this information it seems not reliable, when considered for large displacements and nonlinear material.

3.4.2 Linear Elastic Material

For a linear elastic the constitutive equation is defined in terms of the strain energy function Ψ , the Lamé constants λ and μ and the Cauchy strain tensor given in equation (3.23)

$$\Psi = \mu \boldsymbol{\varepsilon} : \boldsymbol{\varepsilon} + \frac{\lambda}{2} \operatorname{tr}(\boldsymbol{\varepsilon})^2 \quad (3.27)$$

and the Lamé constants are defined as

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \quad (3.28)$$

$$\mu = \frac{E}{2(1+\nu)} \quad (3.29)$$

Where E and ν are Young's modulus and Poisson's ratio, respectively. Young's modulus is a measure of the stiffness of a given material, e.g. an elastic material such as rubber has a very low Young's modulus, whereas a rigid body has an infinite Young's modulus. This giving a relation between how much force needed to deform a specific material. Poisson's ratio is a measure of incompressibility, more formally it is the negative ratio of the lateral strain to the axial strain, e.g. if a body is stretched in one direction it

tends to compress perpendicular to the direction it is being stretched in. Both E and ν are material dependent.

The relation between the first Piola stress tensor, \mathbf{P} , and the deformation gradient, \mathbf{F} can be derived by differentials [11]

$$\delta\boldsymbol{\varepsilon} = \frac{1}{2} (\delta\mathbf{F} + \delta\mathbf{F}^T) = \delta\mathbf{F}_{\text{sym}} \quad (3.30)$$

where \mathbf{F}_{sym} is the symmetric part of a tensor.

$$\boldsymbol{\varepsilon} : \delta\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon} : \delta\mathbf{F}_{\text{sym}} = \text{tr} \left(\boldsymbol{\varepsilon}^T \frac{1}{2} (\delta\mathbf{F} + \delta\mathbf{F}^T) \right) = \text{tr} (\boldsymbol{\varepsilon}^T \delta\mathbf{F}) = \boldsymbol{\varepsilon} : \delta\mathbf{F} \quad (3.31)$$

$$\delta\boldsymbol{\varepsilon} = \mathbf{I} : \delta\mathbf{F}_{\text{sym}} = \text{tr} \left(\mathbf{I}^T \frac{1}{2} (\delta\mathbf{F} + \delta\mathbf{F}^T) \right) = \text{tr} (\mathbf{I}^T \delta\mathbf{F}) = \mathbf{I} : \delta\mathbf{F} \quad (3.32)$$

$$\delta\boldsymbol{\Psi} = 2\mu\boldsymbol{\varepsilon} : \delta\boldsymbol{\varepsilon} + \lambda\text{tr}(\boldsymbol{\varepsilon})\text{tr}(\delta\boldsymbol{\varepsilon}) = [2\mu\boldsymbol{\varepsilon} + \lambda\text{tr}(\boldsymbol{\varepsilon})\mathbf{I}] : \delta\mathbf{F} \quad (3.33)$$

where first Piola stress tensor is

$$\mathbf{P} = \frac{\partial\boldsymbol{\Psi}}{\partial\mathbf{F}} = 2\mu\boldsymbol{\varepsilon} + \lambda\text{tr}(\boldsymbol{\varepsilon})\mathbf{I} \quad (3.34)$$

thus giving us the full relation between stress and strain in a linear elastic in the form of this is a linear function.

3.4.3 Saint Venant-Kirchhoff Material

In the previous subsection was an introduction to a model for a linear elastic. If we proceed with the model, and instead of using the small strain tensor, $\boldsymbol{\varepsilon}$, we replace it with Green strain tensor, \mathbf{E} , in (3.20) for large deformations. This model is called Saint Venant-Kirchhoff material, and is a nonlinear material model. The strain energy function is as follows

$$\boldsymbol{\Psi}(\mathbf{E}) = \mu\mathbf{E} : \mathbf{E} + \frac{\lambda}{2}\text{tr}(\mathbf{E})^2 \quad (3.35)$$

where λ and μ are the Lamé constants shown in equations (3.28) and (3.29).

In the same way as we did the derivation for the relation between \mathbf{P} and \mathbf{F} , we can derive the relation here in the same fashion

$$\delta\mathbf{E} = \frac{1}{2} (\delta\mathbf{F}^T \mathbf{F} + \mathbf{F}^T \delta\mathbf{F}) = [\mathbf{F}^T \delta\mathbf{F}]_{\text{sym}} \quad (3.36)$$

$$\mathbf{E} : \delta\mathbf{E} = \mathbf{E} : [\mathbf{F}^T \delta\mathbf{F}] = [\mathbf{F}\mathbf{E}] : \delta\mathbf{F} \quad (3.37)$$

$$\text{tr}(\delta\mathbf{E}) = \mathbf{I} : [\mathbf{F}^T \delta\mathbf{F}] = \mathbf{F} : \delta\mathbf{F} \quad (3.38)$$

$$\delta\boldsymbol{\Psi} = 2\mu\mathbf{E} : \delta\mathbf{E} + \lambda\text{tr}(\mathbf{E})\text{tr}(\delta\mathbf{E}) = \mathbf{F}[2\mu\mathbf{E} + \lambda\text{tr}(\mathbf{E})\mathbf{I}] : \delta\mathbf{F} \quad (3.39)$$

Thus obtaining the first Piola stress tensor

$$\mathbf{P} = \frac{\partial\boldsymbol{\Psi}}{\partial\mathbf{F}} = \mathbf{F}[2\mu\mathbf{E} + \lambda\text{tr}(\mathbf{E})\mathbf{I}] \quad (3.40)$$

And the second Piola stress tensor will be

$$\mathbf{S} = \frac{\partial \Psi}{\partial \mathbf{E}} = 2\mu \mathbf{E} + \lambda \text{tr}(\mathbf{E}) \mathbf{I} \quad (3.41)$$

This modification of the model is rotationally invariant due to the Green strain tensors properties. This means that the strain energy function is the same, if the body only undergoes a rotation, more formally: A constitutive model is rotationally invariant if and only if

$$\Psi(\mathbf{RF}) = \Psi(\mathbf{F}) = \Psi(\mathbf{FR}) \quad (3.42)$$

for any value of the deformation gradient \mathbf{F} and any rotation matrix $\mathbf{R} : 3 \times 3$. This is true for the St. Venant-Kirchhoff model. Thus the transformations done by this model are assured to have the same strain energy.

The St. Venant-Kirchhoff model shows a fairly good amount of response to large deformations, that the linear elasticity would not be able to handle. In contrast to the Linear material, where the Piola stress is linear, it is now a cubic polynomial.

In figure 2 is depicted the St. Venant-Kirchhoff model, with Cauchy stress as a function of the invariant λ_1 . From this it can be seen that this model can be questionable at some times. E.g. if the material is stretched in the axial direction and remain unstretched in the lateral direction, the Cauchy stress for St. Venant-Kirchhoff is

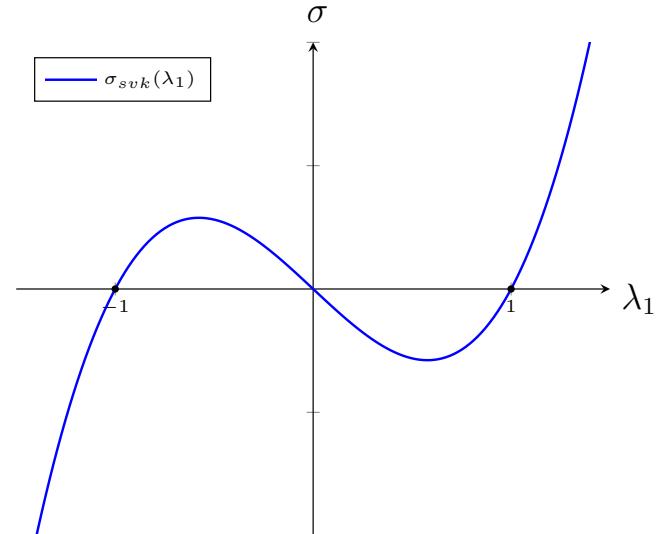
$$\sigma_{11} = \mu (\lambda_1 + \lambda_1^3) + \frac{\lambda}{2} (-\lambda_1 + \lambda_1^3)$$

For a stretch factor, λ_1 , close to 1 would the stress-strain relation behave normally, otherwise the behaviour can be very unpredictable. This is why it only shows a fairly good result to large deformations.

Another distinct property of St. Venant-Kirchhoff is that it is isotropic, which means that its resistance to deformation is the same along all possible directions that the deformation may be applied. [11]

In the section of continuum mechanics there have been laid out the most important theory of the field in question. I have presented the displacement field – also known as deformation map, the deformation gradient and some simple 2D examples thereof, stresses and strains and their relations through two constitutive equations for linear elastic material and nonlinear elastic material.

Figure 2: St. Venant-Kirchhoff model: Cauchy stress as a function of invariants. With $\lambda_2 = \lambda_3$ and Lamé constants $\lambda = \mu = 1$.



4 The Finite Element Method

The finite element method is a method often used in the field of continuum mechanics, where the infinity of nodes in a continuum body is approximated to a finite set of nodes. The set of finite nodes in the continuum body are used to define regions, and these regions are called finite elements. In the finite elements, the geometry and primary variables of the governing equation are approximated. First the governing equation describing the behaviour is recast into a weak integral form, using virtual work and stationary potential energy. Then the finite element approximations are introduced into the integral equations, yielding a differential equation to time integrate. References used in this section [1], [4], [10], [8] and [12].

4.1 Weak Formulation

Rewriting the model problem into volume integral formulation

Cauchy's equation is

$$\rho \ddot{x} = \nabla \cdot \sigma + \bar{b} \quad (4.1)$$

The volume integral is obtained by integration over the material volume, V , while multiplying with a test function \bar{w} . We get

$$\int_V (\rho \ddot{x} - \nabla \cdot \sigma - \bar{b}) \cdot \bar{w} dV = 0 \quad (4.2)$$

where the test function, \bar{w} , is stating a virtual displacement. Which we can split into terms

$$0 = \int_V \rho \ddot{x} \cdot \bar{w} dV - \int_V (\nabla \cdot \sigma) \cdot \bar{w} dV - \int_V \bar{b} \cdot \bar{w} dV \quad (4.3)$$

This can be rewritten by $\nabla \cdot (\sigma \bar{w}) = (\nabla \cdot \sigma) \cdot \bar{w} + \sigma : \nabla \bar{w}^T$ and because σ is symmetric we have this property $\sigma^T = \sigma \Rightarrow \sigma : \Delta \bar{w} = (\sigma : \Delta \bar{w}^T)^T$. We will have

$$0 = \int_V \rho \ddot{x} \cdot \bar{w} dV + \int_V \sigma : \nabla \bar{w} dV - \int_V \nabla \cdot (\sigma \bar{w}) dV - \int_V \bar{b} \cdot \bar{w} dV \quad (4.4)$$

Because of Gauss divergence theorem² we can rewrite the third term using the boundary condition on ∂V_t and obtain

$$\int_V \nabla \cdot (\sigma \bar{w}) dV = \int_{\partial V} (\sigma \bar{w}) \cdot \bar{n} dS \quad (4.5)$$

$$= \int_{\partial V} \bar{w} \cdot (\sigma \bar{n}) dS \quad (4.6)$$

$$= \int_{\partial V} \bar{t} \cdot \bar{w} dS \quad (4.7)$$

$$= \int_{\partial V_t} \bar{t} \cdot \bar{w} dS \quad (4.8)$$

²Gauss Divergence Theorem is stating that we can transform a volume integral into a surface integral over the boundary over the volume by $\int_V (\nabla \cdot \mathbf{F}) dV = \oint_{\partial V} (\mathbf{F} \cdot \bar{n}) dS$, with \mathbf{F} being a continuously differentiable and \bar{n} is the normal to the surface. https://en.wikipedia.org/wiki/Divergence_theorem

where the traction surface \bar{t} is a boundary condition and is applied to the rim of the volume, ∂V_t , and $\bar{t} = 0$ elsewhere.

Which gives us the final volume integral reformulation

$$0 = \int_V \rho \ddot{x} \cdot \bar{w} dV + \int_V \sigma : \nabla \bar{w} dV - \int_{\partial V_t} \bar{t} \cdot \bar{w} dS - \int_V \bar{b} \cdot \bar{w} dV \quad (4.9)$$

If we split this equation into variables we will have a more convenient way of rewriting the terms. This gives us

$$P_\rho = \int_V \rho \ddot{x} \cdot \bar{w} dV \quad (4.10)$$

$$P_e = \int_V \sigma : \nabla \bar{w} dV \quad (4.11)$$

$$P_t = - \int_{\partial V_t} \bar{t} \cdot \bar{w} dS \quad (4.12)$$

$$P_b = - \int_V \bar{b} \cdot \bar{w} dV \quad (4.13)$$

This gives us the full equation

$$0 = P_\rho + P_e + P_t + P_b \quad (4.14)$$

For free floating objects $P_t = 0 \Rightarrow P_\rho + P_e + P_b = 0$.

If we use a viscous damping force as our body force we can rewrite P_b as

$$P_b = - \int_V -c \dot{\bar{x}} \cdot \bar{w} dV = \int_V c \dot{\bar{x}} \cdot \bar{w} dV \quad (4.15)$$

where c is the viscous damping coefficient, $c > 0$.

Virtual elastic energy term is rewritten by the symmetry of Cauchy's stress tensor. This gives us

$$P_e = \int_V \sigma : \frac{1}{2} (\nabla \bar{w} + \nabla \bar{w}^T) dV \quad (4.16)$$

and the infinitesimal strain tensor $\boldsymbol{\varepsilon}$

$$\boldsymbol{\varepsilon} = \frac{1}{2} (\nabla \bar{w} + \nabla \bar{w}^T) \quad (4.17)$$

We therefore define following

$$\boldsymbol{\varepsilon}_{\bar{w}} = \frac{1}{2} (\nabla \bar{w} + \nabla \bar{w}^T) \quad (4.18)$$

And the virtual elastic energy will finally be

$$P_e = \int_V \sigma : \boldsymbol{\varepsilon}_{\bar{w}} dV \quad (4.19)$$

After the changes there has been made our weak form integrals is now as follows

$$P_\rho = \int_V \rho \ddot{\bar{x}} \cdot \bar{w} dV \quad (4.20)$$

$$P_e = \int_V \sigma : \boldsymbol{\varepsilon}_{\bar{w}} dV \quad (4.21)$$

$$P_b = \int_V c \dot{\bar{x}} \cdot \bar{w} dV \quad (4.22)$$

And out mathematical model is as follows

$$0 = P_\rho + P_e + P_b \quad (4.23)$$

4.2 Shape Functions

The finite elements used in this project is a tetrahedral structure.

The assemble process is defined as

$$\int_V (\dots) dV = \sum_e \int_{V^e} (\dots) dV \quad (4.24)$$

where the left side is the volume integral over the whole body and the right side is the sum of the volume integral of one single tetrahedron element.

This means that you can without loss of generality simply derive an expression for a single element and thereafter assemble for all the elements in the tetrahedral structure.

Consider a point \bar{X} and displacement \bar{u} inside a tetrahedron, with nodal labels $e \equiv \{i, j, k, m\}$. Then the values are interpolated using linear shape functions by

$$\bar{X} = N_i \bar{X}_i + N_j \bar{X}_j + N_k \bar{X}_k + N_m \bar{X}_m \quad (4.25)$$

$$\bar{u} = N_i \bar{u}_i + N_j \bar{u}_j + N_k \bar{u}_k + N_m \bar{u}_m \quad (4.26)$$

where N_a for $a \in e$ are volume coordinates with respect to the e 'th tetrahedron.

The interpolation weights are computed as

$$N_i = \frac{V_{jkm}}{V^e} \quad (4.27)$$

$$= \frac{(\bar{X} - \bar{X}_j) \cdot ((\bar{X}_k - \bar{X}_j) \times (\bar{X}_m - \bar{X}_k))}{(\bar{X}_m - \bar{X}_i) \cdot ((\bar{X}_j - \bar{X}_i) \times (\bar{X}_k - \bar{X}_i))} \quad (4.28)$$

where V^e is the material volume corresponding to one tetrahedron element and V_{jkm} is the volume of the sub tetrahedron spanned by \bar{X} with face nodes j, k, m . Material gradients of weights is given as following by the above mentioned computation

$$\nabla_0 N_i = \frac{(\bar{X}_k - \bar{X}_j) \times (\bar{X}_m - \bar{X}_k)}{6V^e}. \quad (4.29)$$

Interpolating the corresponding discrete nodal values from the computational mesh leads to following approximation

$$\bar{u} \approx \sum_a N_a \bar{u}_a = \mathbf{N}^e \tilde{\bar{u}}^e \quad (4.30)$$

This is also true for the test function \bar{w} and the spatial positions \bar{x} and $\mathbf{N}^e = [N_i \mathbf{I} \quad N_j \mathbf{I} \quad N_k \mathbf{I} \quad N_m \mathbf{I}]$.

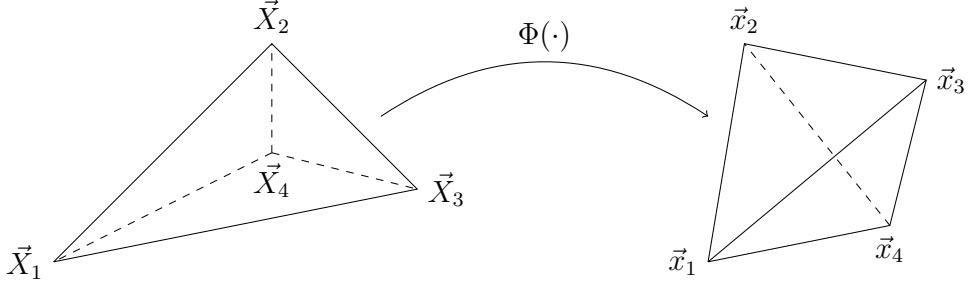


Figure 3: A Linear Tetrahedron³ deformed by a function $\Phi(\cdot)$.

4.3 Nonlinear Finite Element Method

The model obtained in the weak reformulation in terms of material volume is rewritten in terms of the Piola stress tensor, rather than the Cauchy stress tensor. This gives us following for the elastic stress term

$$P_e = \int_V \sigma : \nabla \bar{w} dV = \int_V \mathbf{P} \mathbf{F}^T : \nabla \bar{w} dV \quad (4.31)$$

$$= \int_V \mathbf{P} \mathbf{F}^T : \nabla_0 \bar{w} \mathbf{F}^{-1} dV \quad (4.32)$$

$$= \int_V \mathbf{P} : \nabla_0 \bar{w} dV \quad (4.33)$$

From the shape function discretization we have the approximation in equation (4.30), which means that

$$\nabla_0 \bar{w} = \sum_a \tilde{\bar{w}}_a \otimes \nabla_0 N_a \quad (4.34)$$

and the final expression for the elastic term is obtained

$$P_e = \int_V \mathbf{P} : \left(\sum_a \tilde{\bar{w}}_a \otimes \nabla_0 N_a \right) dV \quad (4.35)$$

$$= \sum_a \left(\tilde{\bar{w}}_a \cdot \int_V \mathbf{P} \nabla_0 N_a dV \right) \quad (4.36)$$

Then the mathematical function

$$P_\rho + P_e + P_b = 0 \quad (4.37)$$

must hold for all $\tilde{\bar{w}}_a$ and by substituting the shape functions we obtain a final equation for a single tetrahedron element

$$\mathbf{M}^e \ddot{\bar{x}}^e + \mathbf{C}^e \dot{\bar{x}}^e + \bar{k}^e (\tilde{\bar{u}}^e) = \bar{f}^e \quad (4.38)$$

³A solid with four vertices and four triangular faces.

where \bar{f}^e is a constant body force and the terms are defined as

$$\mathbf{M}^e = \int_{V^e} \rho_0 (\mathbf{N}^e)^T \mathbf{N}^e dV \quad (4.39)$$

$$\mathbf{C}^e = \int_{V^e} c_0 (\mathbf{N}^e)^T \mathbf{N}^e dV \quad (4.40)$$

$$\bar{k}_a^e(\tilde{u}^e) = \int_{V^e} \mathbf{P} \nabla_0 N_a dV \quad (4.41)$$

$$\bar{k}^e = \begin{bmatrix} \bar{k}_i^e \\ \bar{k}_j^e \\ \bar{k}_k^e \\ \bar{k}_m^e \end{bmatrix} \quad (4.42)$$

When the assemble process has been done, we have the final differential equation to time integrate

$$\mathbf{M} \ddot{\bar{x}} + \mathbf{C} \dot{\bar{x}} + \bar{k}(\tilde{u}) = \bar{f}. \quad (4.43)$$

During this section there has been an introduction to finite element method as a small and concise walk-through the process of assemble a continuum body of tetrahedra.

5 Discretization

The result of the previous section is a second order differential equation (4.43) of which in this section will be discretized by an implicit Euler time integration with the help of finite difference approximations. We will leave the tilde notation to the finite element method and thus referring to the differential equation as

$$\mathbf{M} \ddot{\bar{x}} + \mathbf{C} \dot{\bar{x}} + \bar{k}(\bar{u}) = \bar{f} \quad (5.1)$$

For stiff materials the update rules requires a very small time step to be stable, thus using an fully implicit scheme to gain an unconditionally stable method. Hence, the discretization method will be a fully implicit backward Euler scheme. The section will involve a detailed derivation of the equation in (5.1) from a block matrix system into a linear system of equations. With knowledge obtained from [4] and [11].

A first order finite difference approximation by the use of forward Euler is introduced for the stiffness tensor. [6]

5.1 Implicit Time Integration by Backward Euler

If we start by rewriting the second order differential equation into a system of coupled first order differential equations, we achieve

$$\dot{\bar{x}} = \bar{v} \quad (5.2a)$$

$$\mathbf{M} \dot{\bar{v}} + \mathbf{C} \bar{v} + \bar{k}(\bar{x} - \bar{X}) = \bar{f} \quad (5.2b)$$

Then using the backward Euler scheme to discretize the positions and velocities at $\bar{x}^{t+\Delta t}$ and $\bar{v}^{t+\Delta t}$, respectively

$$\bar{x}^{t+\Delta t} = \bar{x}^t + \Delta t \bar{v}^{t+\Delta t} \quad (5.3a)$$

$$\bar{v}^{t+\Delta t} = \bar{v}^t + \Delta t \mathbf{M}^{-1} (\bar{f} - \mathbf{C} \bar{v}^{t+\Delta t} - \bar{k} (\bar{x} - \bar{X})) \quad (5.3b)$$

The above system is nonlinear due to equation (5.3b). By simplifying the nonlinear system above, we achieve the system

$$\bar{x}^{t+\Delta t} - \bar{x}^t - \Delta t \bar{v}^{t+\Delta t} = \bar{0} \quad (5.4a)$$

$$\mathbf{M} (\bar{v}^{t+\Delta t} - \bar{v}^t) + \Delta t (\mathbf{C} \bar{v}^{t+\Delta t} + \bar{k} (\bar{x}^{t+\Delta t} - \bar{X}) - \bar{f}) = \bar{0} \quad (5.5)$$

We define

$$\bar{\Phi}_x (\bar{x}, \bar{v}) := \bar{x} - \bar{x}^t - \Delta t \bar{v} \quad (5.6a)$$

$$\bar{\Phi}_v (\bar{x}, \bar{v}) := (\mathbf{M} + \Delta t \mathbf{C}) \bar{v} - \mathbf{M} \bar{v}^t - \Delta t \bar{f} + \Delta t \bar{k} (\bar{x} - \bar{X}) \quad (5.6b)$$

which we can put together as

$$\bar{\Phi} (\bar{x}, \bar{v}) = \begin{bmatrix} \bar{\Phi}_x (\bar{x}, \bar{v}) \\ \bar{\Phi}_v (\bar{x}, \bar{v}) \end{bmatrix} = \bar{0} \quad (5.7)$$

Where the system above is equal to a nonlinear root search problem [6] and therefore it is possible to solve the system at the k 'th iteration of the Newton method by making a first order approximation around the k 'th values.

$$\begin{aligned} \bar{\Phi} (\bar{x}^{t+\Delta t}, \bar{v}^{t+\Delta t}) &\approx \bar{\Phi} (\bar{x}_{(k)}^{t+\Delta t}, \bar{v}_{(k)}^{t+\Delta t}) + \frac{\partial \bar{\Phi} (\bar{x}_{(k)}^{t+\Delta t}, \bar{v}_{(k)}^{t+\Delta t})}{\partial \bar{x}_{(k)}^{t+\Delta t}} \Delta \bar{x}_{(k)}^{t+\Delta t} \\ &\quad + \frac{\partial \bar{\Phi} (\bar{x}_{(k)}^{t+\Delta t}, \bar{v}_{(k)}^{t+\Delta t})}{\partial \bar{v}_{(k)}^{t+\Delta t}} \Delta \bar{v}_{(k)}^{t+\Delta t} = \bar{0} \end{aligned} \quad (5.8b)$$

Where the subparts of the equation above, including the partial derivatives is

$$\bar{\Phi} (\bar{x}_{(k)}^{t+\Delta t}, \bar{v}_{(k)}^{t+\Delta t}) = \begin{bmatrix} \bar{x}_{(k)}^{t+\Delta t} - \bar{x}^t - \Delta t \bar{v}_{(k)}^{t+\Delta t} \\ (\mathbf{M} + \Delta t \mathbf{C}) \bar{v}_{(k)}^{t+\Delta t} - \mathbf{M} \bar{v}^t - \Delta t \bar{f} + \Delta t \bar{k} (\bar{x}_{(k)}^{t+\Delta t} - \bar{X}) \end{bmatrix} \quad (5.9a)$$

$$\frac{\partial \bar{\Phi} (\bar{x}_{(k)}^{t+\Delta t}, \bar{v}_{(k)}^{t+\Delta t})}{\partial \bar{x}_{(k)}^{t+\Delta t}} = \begin{bmatrix} \mathbf{I} \\ \frac{\partial \bar{k} (\bar{x}_{(k)}^{t+\Delta t} - \bar{X})}{\partial \bar{x}_{(k)}^{t+\Delta t}} \Delta t \end{bmatrix} \quad (5.9b)$$

$$\frac{\partial \bar{\Phi} (\bar{x}_{(k)}^{t+\Delta t}, \bar{v}_{(k)}^{t+\Delta t})}{\partial \bar{v}_{(k)}^{t+\Delta t}} = \begin{bmatrix} -\Delta t \\ \mathbf{C} \Delta t + \mathbf{M} \end{bmatrix} \quad (5.9c)$$

And the full system of (5.8b) will be

$$\begin{aligned}\bar{\Phi}(\bar{x}^{t+\Delta t}, \bar{v}^{t+\Delta t}) &\approx \left[\begin{array}{c} \bar{x}_{(k)}^{t+\Delta t} - \bar{x}^t - \Delta t \bar{v}_{(k)}^{t+\Delta t} \\ (\mathbf{M} + \Delta t \mathbf{C}) \bar{v}_{(k)}^{t+\Delta t} - \mathbf{M} \bar{v}^t - \Delta t \bar{f} + \Delta t \bar{k}(\bar{x}_{(k)}^{t+\Delta t} - \bar{X}) \end{array} \right] \\ &+ \left[\begin{array}{c} \mathbf{I} \\ \frac{\partial \bar{k}(\bar{x}_{(k)}^{t+\Delta t} - \bar{X})}{\partial \bar{x}_{(k)}^{t+\Delta t}} \Delta t \end{array} \right] \Delta \bar{x}_{(k)}^{t+\Delta t} \\ &+ \left[\begin{array}{c} -\Delta t \\ \mathbf{C} \Delta t + \mathbf{M} \end{array} \right] \Delta \bar{v}_{(k)}^{t+\Delta t} = \bar{0}\end{aligned}\quad (5.10)$$

which is then solved for the k 'th Newton direction, that is used in the k 'th Newton update

$$\bar{x}_{(k+1)}^{t+\Delta t} = \bar{x}_{(k)}^{t+\Delta t} + \Delta \bar{x}_{(k)} \quad (5.11)$$

$$\bar{v}_{(k+1)}^{t+\Delta t} = \bar{v}_{(k)}^{t+\Delta t} + \Delta \bar{v}_{(k)} \quad (5.12)$$

Unless it is pointed out explicit, we will without loss of generality use following \bar{x} to describe $\bar{x}_{(k)}^{t+\Delta t}$ and \bar{v} to describe $\bar{v}_{(k)}^{t+\Delta t}$. The same goes for $\bar{\Phi}_x := \bar{\Phi}_x(\bar{x}, \bar{v})$ and $\bar{\Phi}_v := \bar{\Phi}_v(\bar{x}, \bar{v})$. Equations (5.9b) and (5.9c) on closed form solution for the Jacobian : $\mathbf{J} \Delta \bar{x} = \bar{b}$ is therefore

$$\bar{\Phi}_x = \begin{bmatrix} \mathbf{I} \\ \Delta t \frac{\partial \bar{k}}{\partial \bar{u}} \end{bmatrix} \quad (5.13a)$$

$$\bar{\Phi}_v = \begin{bmatrix} -\Delta t \mathbf{I} \\ \mathbf{M} + \Delta t \mathbf{C} \end{bmatrix} \quad (5.13b)$$

the use of the closed-form Jacobians is to write an expression for the nonlinear compact Newton system of block matrices as follows

$$\begin{bmatrix} \mathbf{I} & -\Delta t \mathbf{I} \\ \Delta t \frac{\partial \bar{k}}{\partial \bar{u}} & (\mathbf{M} + \Delta t \mathbf{C}) \end{bmatrix} \begin{bmatrix} \Delta \bar{x} \\ \Delta \bar{v} \end{bmatrix} = \begin{bmatrix} -\bar{\Phi}_x \\ -\bar{\Phi}_v \end{bmatrix} \quad (5.14)$$

To obtain a linear system of ordinary differential equations, one must linearize the block system of equations in (5.14).

From the second row in the system, we achieve the following by solving for $\Delta \bar{v}$

$$\Delta t \frac{\partial \bar{k}}{\partial \bar{u}} \Delta \bar{x} (\mathbf{M} + \Delta t \mathbf{C}) \Delta \bar{v} = -\bar{\Phi}_v \quad (5.15)$$

$$\Delta \bar{v} = (\mathbf{M} + \Delta t \mathbf{C})^{-1} \left(-\bar{\Phi}_v - \Delta t \frac{\partial \bar{k}}{\partial \bar{u}} \Delta \bar{x} \right) \quad (5.16)$$

In the first row, we then substitute $\Delta\bar{v}$, by the result obtained in (5.16)

$$\Delta\bar{x} - \Delta t \Delta\bar{v} = -\bar{\Phi}_x \quad (5.17a)$$

$$\Delta\bar{x} - \Delta t \left((\mathbf{M} + \Delta t \mathbf{C})^{-1} \left(-\bar{\Phi}_v - \Delta t \frac{\partial \bar{k}}{\partial \bar{u}} \Delta x \right) \right) = -\bar{\Phi}_x \quad (5.17b)$$

$$\Delta\bar{x} - \Delta t \left(-(\mathbf{M} + \Delta t \mathbf{C})^{-1} \bar{\Phi}_v - (\mathbf{M} + \Delta t \mathbf{C})^{-1} \Delta t \frac{\partial \bar{k}}{\partial \bar{u}} \Delta\bar{x} \right) = -\bar{\Phi}_x \quad (5.17c)$$

$$\Delta\bar{x} + (\mathbf{M} + \Delta t \mathbf{C})^{-1} \Delta t \bar{\Phi}_v + (\mathbf{M} + \Delta t \mathbf{C})^{-1} \Delta t^2 \frac{\partial \bar{k}}{\partial \bar{u}} \Delta\bar{x} = -\bar{\Phi}_x \quad (5.17d)$$

$$\Delta\bar{x} + (\mathbf{M} + \Delta t \mathbf{C})^{-1} \left(\Delta t \bar{\Phi}_v + \Delta t^2 \frac{\partial \bar{k}}{\partial \bar{u}} \Delta\bar{x} \right) = -\bar{\Phi}_x \quad (5.17e)$$

Multiplying with $(\mathbf{M} + \Delta t \mathbf{C})$ we get

$$(\mathbf{M} + \Delta t \mathbf{C}) \Delta\bar{x} + \Delta t \bar{\Phi}_v + \Delta t^2 \frac{\partial \bar{k}}{\partial \bar{u}} \Delta\bar{x} = -(\mathbf{M} + \Delta t \mathbf{C}) \bar{\Phi}_x \quad (5.18a)$$

$$\left(\mathbf{M} + \Delta t \mathbf{C} + \Delta t^2 \frac{\partial \bar{k}}{\partial \bar{u}} \right) \Delta\bar{x} = -\Delta t \bar{\Phi}_v - (\mathbf{M} + \Delta t \mathbf{C}) \bar{\Phi}_x \quad (5.18b)$$

Where the last line (5.18b) is the final result of the linearization, which is a linear system of ordinary differential equations and corresponds to the matrix-vector product $\mathbf{J}\Delta\bar{x} = \bar{b}$.

This equation is symmetric and positive definite. Therefore it is perfectly designed for an iterative solver such as preconditioned conjugate gradient, for solving for $\Delta\bar{x}$ and afterwards updating the velocities by $\bar{v}_{(k+1)}^{t+\Delta t} = \bar{v}_{(k)}^{t+\Delta t} + \frac{1}{\Delta t} \Delta\bar{x}$.

If one chooses to work with the nonlinear system in equation (5.14), an iterative solver such as Generalized minimal residuals is a good choice as the system only need to be symmetric.

5.2 Approximation of the Stiffness Tensor

The stiffness tensor is as mentioned before the elastic forces differentiated in every entry. This can quickly become extremely expensive if you look at the computational cost. The larger and stiffer the object is, the more expensive the operation will be. Hence, we introduce a finite difference approximation of the stiffness tensor. An finite difference approximation do not involve differentiating in every entry, why this could become less expensive and therefore advantageous. The approximation is only build around the elastic forces with respect to a present set of points and a displacement.

With a view of a matrix-free solution – which will be explained in section 6.1 – where we only need a relation between the Jacobian and the vector we are solving the system for, which gives us a reason to believe that the approximation for the stiffness tensor could be very beneficial.

A first order approximation with a forward Euler is

$$\mathbf{K}\Delta\bar{x} = \frac{\partial \bar{k}(\bar{u})}{\partial \bar{u}} \Delta x = \frac{\bar{k}(\bar{u} + \Delta\bar{x}) - \bar{k}(\bar{u})}{\|\Delta\bar{x}\|_2} + \mathcal{O}(\Delta\bar{x}) \quad (5.19)$$

When removing the big-O term, $\mathcal{O}(\Delta\bar{x})$, we achieve the approximation of $\mathbf{K}\Delta\bar{x}$.

If we substitute this into reduced system in equation (5.18b) we obtain

$$(\mathbf{M} + \Delta t \mathbf{C}) \Delta\bar{x} + \Delta t^2 \frac{\bar{k}(\bar{u} + \Delta\bar{x}) - \bar{k}(\bar{u})}{\|\Delta\bar{x}\|_2} = -\Delta t \bar{\Phi}_v - (\mathbf{M} + \Delta t \mathbf{C}) \bar{\Phi}_x \quad (5.20)$$

This is the system we are going to work with, throughout the project. An I will define this as $\mathbf{A}\Delta\bar{x} = \bar{b}$ for later usage.

In the section of discretization there have been a detailed derivation of the mathematical equation in hand, from a nonlinear block matrix system to a linear system of equations. Further there have been an introduction to a first order forward Euler finite difference approximation of the stiffness tensor, idealized by a matrix free solver.

6 Implementation

In order to visualize all the theory presented during the paper, an implementation were performed. I have used a framework designed by my advisor, Kenny Erleben, as a foundation for my work. I have modified some parts to make it compatible with the aspects that define the view of my paper.

This is going to be used for the experiments presented in the next section.

6.1 A Matrix-Free Solution

As mentioned earlier, the ordinary differential equation shown in (5.18b) is perfectly designed for using preconditioned conjugate gradient(PCG) as an iterative solver for solving the system for $\Delta\bar{x}$, and then updating $\Delta\bar{v}$.

This is also applicable for the block-matrix system in equation (5.14), but not by PCG and instead by using generalized minimal residual method(GMRES).

For both cases is MATLAB allowing us to do this in a matrix-free fashion, meaning that we do not need to explicit know the matrix $\mathbf{J} := \mathbf{M} + \Delta t \mathbf{C} + \Delta t^2 \mathbf{K}$, we need only an expression for the vector product of $\mathbf{J}\bar{v}$. This is beneficial as a result of getting rid of explicitly solving the Jacobian system and storing the system in every evaluation. Which of course is only worsened when solving the matrix-block system, which is nonlinear.

6.2 Computing the Stiffness Tensor

When computing the stiffness tensor in MATLAB it is a possibility to make use of the symbolic capabilities, and highly recommended when working with tensors, as it often requires 4 or more nested for loops.

One of the symbolic properties is differentiation, from which we can shape the first Piola stress tensor as seen in table 1. Before doing this, you need to declare which

variables are used as symbols, then define your strain tensor and deformation gradient, and a constitutive equation in terms of Ψ .

Then the simple code below will give you a MATLAB function, which computes the first Piola stress tensor, from the relation of the deformation gradient and constitutive equation you provided.

Table 1: Symbolic Differentiation of The First Piola Stress Tensor in MATLAB

```

1 % First Piola--Kirchhoff stress tensor
2 P = sym('TEMP')*ones(3,3);
3 for i=1:3
4     for j=1:3
5         P(i,j) = diff(psi F(i,j));
6     end
7 end
8 P = simplify(P)
9 matlabFunction(P, 'file', 'compute_P.m')
```

This is highly advantageous, and can be done for every constitutive models.

As for the stiffness tensor, it can be constructed by the use of the material elasticity tensor $\mathbf{A} = \frac{\partial \mathbf{P}}{\partial \mathbf{F}}$.

Table 2: Symbolic Computation of The Stiffness Tensor in MATLAB, with \mathbf{A} being the Material Elasticity Tensor

```

1 % Tangent stiffness element
2 syms Ga1 Ga2 Ga3 real;
3 syms Gb1 Gb2 Gb3 real;
4 Ga = [Ga1; Ga2; Ga3];
5 Gb = [Gb1; Gb2; Gb3];
6
7 Kab = sym('TEMP')*ones(3,3);
8 Kab(:,:) = 0;
9 for i=1:3
10     for k=1:3
11         for j=1:3
12             for m=1:3
13                 r = (i-1)*3 + k;
14                 c = (j-1)*3 + m;
15                 Kab( i, k ) = Kab( i, k ) + ...
16                         A(r,c)*Gb(m)*Ga(j);
17             end
18         end
19     end
20 end
21 Kab = simplify(Kab)
22 matlabFunction(Kab, 'file', 'compute_K.m');
```

which results in a closed form derivation of the stiffness tensor.

The symbolic code in MATLAB is drafted by my advisor, Kenny Erleben in [4]

7 Experiments

It is known from previous work such as [9] that a numerical approximation of a the stiffness tensor is possible when working with non-physically based model. Thus the question remaining: is it also applicable when working with physically based model with real parameters.

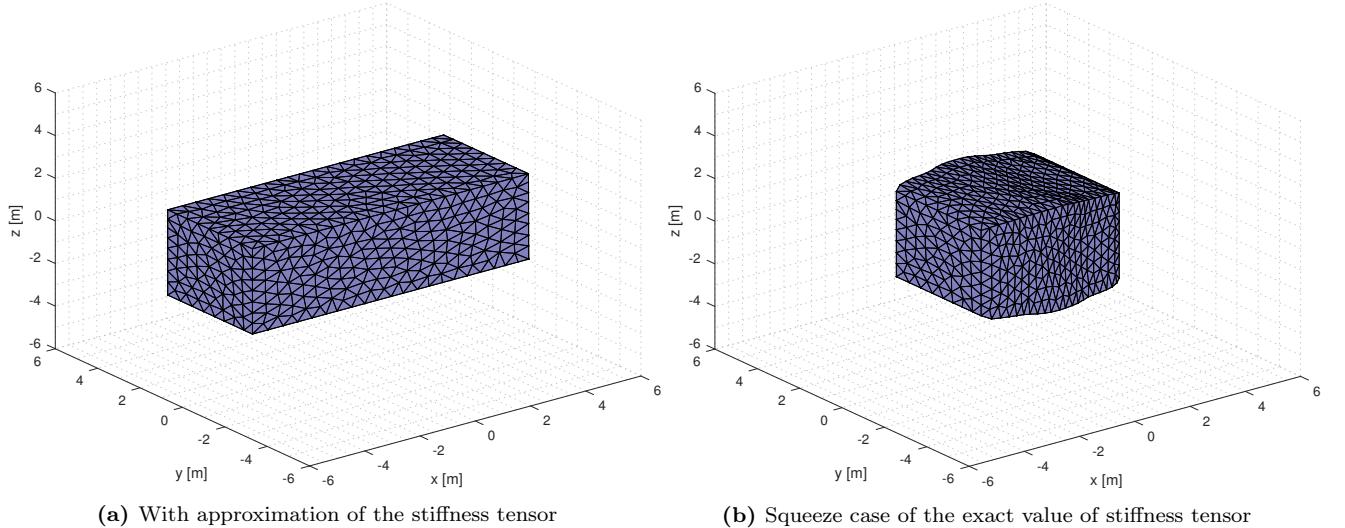


Figure 4: Figure 4a showing the squeeze case with the approximation of the deformation tensor. Figure 4b showing the squeeze case with the exact value of the stiffness tensor. Both is mesh number 5.

The figures shown above are two simulations with identical parameter settings besides the choice of how to compute the stiffness tensor. In figure 4a is a simulation that corresponds to the reference frame, except it is not. It is a simulation with the application of a first order finite difference approximation of the stiffness tensor. In figure 4b a simulation is shown, which initial frame is equal to what is shown in figure 4a. This is done with the application of the exact version of the stiffness tensor, and this shows a great deal of deformation, which is what we wished for when applying the approximation. Simulation time is the same for both instances, but the result is not.

By looking at both a deformed body, and an undeformed body(that is supposed to be deformed), it is easy to reach at the conclusion, that there is something that did not fully coincide, and that the approximation of the stiffness tensor is not behaving as expected. The reason why the approximation if not behaving as expected is hard to come by and can lie within many aspects of the project. With a lot of questions come a lot of experiments.

Due to the fact that we know that is it possible to approximate the stiffness tensor for a non-physically based model, it can be a suggested that it just do not comply with real-valued stiffnesses. Hence, an inspection of what might have gone wrong.

7.1 Properties for Using an Iterative Solver

Perhaps the application of the finite difference approximation are destroying the symmetry and positive definiteness that is needed when using the iterative solver, preconditioned conjugate gradient.

The Jacobian \mathbf{J} is symmetric and positive definite if and only if both $\mathbf{J} = \mathbf{J}^T$ and $\bar{v}^T \mathbf{J} \bar{v} > 0$ for all non-zero vectors \bar{v} , [6] where $\mathbf{J} := \mathbf{M} + \Delta t \mathbf{C} + \Delta t^2 \mathbf{K}$ from the ODE obtained in equation (5.18b) in section 5. I simply checked this in MATLAB and the result showed that \mathbf{J} is both symmetric and positive definite.

This must also apply for when we use the approximation of \mathbf{K} instead of the exact value, but when we use a matrix-free computation to evaluate $\mathbf{A}(\bar{v})$,

$$\mathbf{A}(\bar{v}) = (\mathbf{M} + \Delta t \mathbf{C}) \bar{v} + \Delta t^2 \left(\frac{\bar{k}(\bar{x} + \bar{v}) - \bar{k}(\bar{x})}{\|\bar{v}\|_2} \right) \quad (7.1)$$

which solution is a vector. Yet it is still possible to check whether or not the positive definiteness is preserved, and this is also true cf. MATLAB. Although we cannot check if it is symmetric, the same way we did the Jacobian, we can check whether or not $\bar{v}^T \mathbf{A} \bar{u}$ is equivalent with $\bar{v}^T \mathbf{J} \bar{u}$, for two user determined vector \bar{v} and \bar{u} . Thus the problem stated is

$$\bar{v}^T \mathbf{A}(\bar{u}) \stackrel{?}{=} \bar{v}^T \mathbf{J} \bar{u} \quad (7.2)$$

Table 3: Table of error scale, when checking the symmetry of the matrix-free product $\mathbf{A}(\bar{v})$ against the symmetry of the Jacobian \mathbf{J} , with different test vectors, and a tolerance of 10^{-5} .

2500 Test Vectors	St. Venant-Kirchhoff Material		Linear Isotropic Material	
	Stepsize h	Error [%]	Stepsize h	Error [%]
Unit Vectors	$h = 10^{-2}$	err = 10.7%	$h = 10^{-2}$	err = 10.6%
	$h = 10^{-3}$	err = 10.7%	$h = 10^{-3}$	err = 10.6%
	$h = 10^{-5}$	err = 9.5%	$h = 10^{-5}$	err = 9.2%
Normally Distributed Random Generated Vectors with $\mu = 0$ and $\sigma = 1$	$h = 10^{-2}$	err = 100%	$h = 10^{-2}$	err = 100%
	$h = 10^{-3}$	err = 100%	$h = 10^{-3}$	err = 100%
	$h = 10^{-5}$	err = 100%	$h = 10^{-5}$	err = 99.96%
Normally Distributed Random Generated Vectors with $\mu = \mu_{real}$ and $\sigma = \sigma_{real}$	$h = 10^{-2}$	err = 100%	$h = 10^{-2}$	err = 100%
	$h = 10^{-3}$	err = 98%	$h = 10^{-3}$	err = 98.64%
	$h = 10^{-5}$	err = 0%	$h = 10^{-5}$	err = 0%

Table 3 is displaying results from the computation of the question in hand – is \mathbf{A} symmetric?

The results of the table are computed with three sets of different test vectors, including unit vectors, and two kinds of normally distributed random generated data. One set of vectors generated by the build-in `randn()` in MATLAB used a mean, $\mu = 0$ and a standard deviation, $\sigma = 1$, which means that 68% of the data lies within $[-1, 1]$. The other set of random vectors computed by retrieving displacement vectors from a simulation with the exact value of \mathbf{K} , hereafter finding the mean and standard deviation

to compute 2500 vectors corresponding to “real” displacements, by following formula

$$test_vectors_{real} := \bar{t} = \mu_{real} \cdot \text{randn}() + \sigma_{real} \quad (7.3)$$

All the error percentages is found from the absolute difference of the left and the right side of equation (7.2), with a tolerance of 10^{-5} .

Table 3 implies that the best combination of test vectors and step size h , is the random generated vectors, \bar{t} and a step size equal to 10^{-5} , for a tolerance 10^{-5} . This is true for both material models.

The error percentage for random generation vectors, denoted \bar{r} , as displacements, could indicate that the finite difference approximation for both material models cannot handle the diversity that follows from the use of random vectors. Whereas both models seems to deal with unit vectors quite well, unfortunately are the real world displacements rarely comparable to unit vectors.

7.2 Order Of Approximation

Is it an acceptable assumption, that it is sufficient to use a first order finite difference method to approximate the stiffness tensor? For it to be necessary to do a second order or even higher orders of finite difference approximations, the elastic forces should be non-linear. E.g. if the elastic forces are linear, that means we are able to find an approximate to a point by looking at the tangent line through two points, that we evaluate. This corresponds to a first order finite difference. On the other hand, if the elastic forces are non-linear we would need to evaluate three or more points to find an approximate to the original point, which corresponds to a second or higher order finite difference method. If the elastic forces are nonlinear this would reveal why the approximation of the stiffness tensor not seem to converge to a result worth simulating.

In order to analyse the order of the finite difference method needed to properly approximate the stiffness tensor, I did a test in MATLAB with the elastic forces \bar{k} as a function of $\tau \in [0, 10]$, scaling a random generated vector \bar{t} as in equation (7.3) that represents a real displacement. The vector \bar{t} is an example of a deformation we want to achieve with the approximation. The elastic forces as a function of some scaling τ . We define

$$\bar{\kappa}(\tau_i) := \bar{k}(\bar{x} + \tau_i \bar{t}) \quad \tau_i \in [0, 10] \quad (7.4)$$

And as a consequence of τ ’s domain we have $\bar{k}(\bar{x} + \tau \bar{t}) \rightarrow \bar{k}(\bar{x})$ as $\tau \rightarrow 0$, this is the relation we wish for to be linear in order for us to make use of a first order approximation.

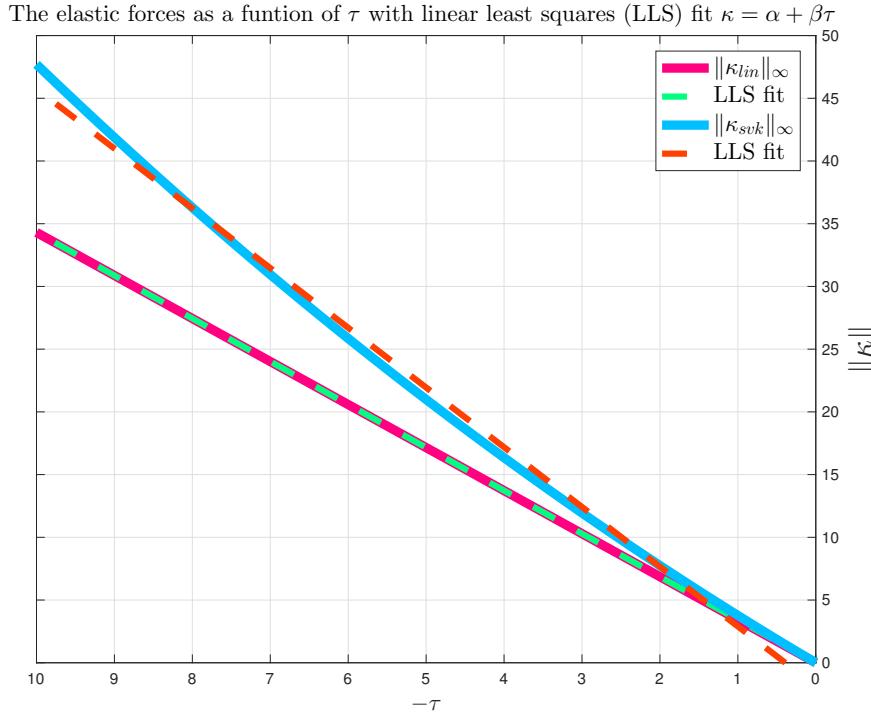


Figure 5: The elastic forces as a function of τ , for St. Venant-Kirchhoff and Linear Elastic material, respectively. Both fitted with a Linear Least Squares model (LLS) with $\kappa = \alpha + \beta\tau$.

When looking at the figure 5, seen with the naked eye, the lines look dead straight, and it seems most likely that the elastic forces are linear – for both models. Although, in order to be completely sure, the data from both models were fitted by a linear least squares model.[13]

The weights α and β are each found from

$$\alpha = \frac{\sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i}{N \sum x_i^2 - (\sum x_i)^2} \quad (7.5)$$

$$\beta = \frac{N \sum x_i y_i - \sum x_i \sum y_i}{N \sum x_i^2 - (\sum x_i)^2} \quad (7.6)$$

where N is the number of points, $x := \tau$ and $y := \kappa$. After performing the linear least squares fit, it could be doubted whether or not that κ_{svk} is linear. This could also just be a bad fit. There is bound to be uncertainties, when working with fits and especially random generated data.

The scale of κ is highly dependent by the value of Young's modulus, E , and if the order in magnitude of E is 10^{15} , so is the order in magnitude of κ . This is where the two models of St. Venant-Kirchhoff and Linear Elastic, seems to differ from one another. It may be that the stiffness of a given continuum, is more significant in the elastic forces of an St. Venant-Kirchhoff material, compared to a Linear elastic material.

This means that the usage of a second order finite difference approximation would not be giving us any advantage over a first order finite difference approximation.

7.3 Crossvalidation of Parameters

The crossvalidation of the parameters involved in the project is namely of the most important parameters such as Young's modulus, E , Poisson's ratio, ν , mesh size, L , and the time step size, h . One could also choose to do a crossvalidation for other parameters, e.g. the density, ρ , but seeing that the mass matrix is constant it should not have an impact.

The crossvalidation is giving us a view on how the finite difference approximation is behaving compared to the exact version off the stiffness tensor \mathbf{K} , for every value of h and the parameter, $P := \{E, \nu, L\}$.

The motivation for crossvalidating the parameters is in hoping to find a feasible area of which would cause the simulation with the approximation to work.

This experiment is involving the absolute error of the approximation and the exact version for 30 test vectors, \bar{v} , where $\bar{v} = \{\bar{e}, \bar{r}, \bar{t}\}$, for every single value of 100 values for $h \in [10^{-2}, 10^{-30}]$, $E \in [10^2, 10^{15}]$, $\nu \in [0, \frac{1}{2}]$ and for eight values, $L = \{1, 2, 3, 4, 5, 6, 7, 8\}$

All the crossvalidations are computed by implementing this pseudocode

Algorithm 1 Crossvalidation of $h \times P$

```

1: procedure CROSSVAL( $P, \bar{v}$ )
2:    $h \leftarrow [10^{-2}, 10^{-3}]$ 
3:   for  $i$  in  $h$  do
4:     for  $j$  in  $P$  do
5:        $\bar{k} \leftarrow \text{compute\_elastic\_force\_elements}(h_i, P_j)$ 
6:        $\mathbf{K} \leftarrow \text{compute\_stiffness\_elements}(h_i, P_j)$ 
7:       for  $\ell$  in  $\bar{v}$  do
8:         approx  $\leftarrow \text{compute\_approx}(h_i, P_j, \bar{v}_\ell)$ 
9:          $\bar{K}_{\text{approx}} \leftarrow h_i^2 \cdot \text{approx}$ 
10:         $\bar{K}_{\text{exact}} \leftarrow h_i^2 \cdot \mathbf{K} \bar{v}_\ell$ 
11:         $\bar{\varepsilon}_\ell \leftarrow \|\bar{K}_{\text{approx}} - \bar{K}_{\text{exact}}\|_\infty$ 
12:         $\delta_{i,j} \leftarrow \|\bar{\varepsilon}\|_\infty$ 
return  $\delta$ 

```

where $P = \{E, \nu, L\}$ $\bar{v} = \{\bar{e}, \bar{r}, \bar{t}\}$ The absolute error for the first order approximation, is more explicitly

$$\varepsilon = \left\| h^2 \left(\frac{\bar{k}(\bar{x} + \bar{v}) - \bar{k}(\bar{x})}{\|\bar{v}\|_2} \right) - h^2 \mathbf{K} \bar{v} \right\|_\infty. \quad (7.7)$$

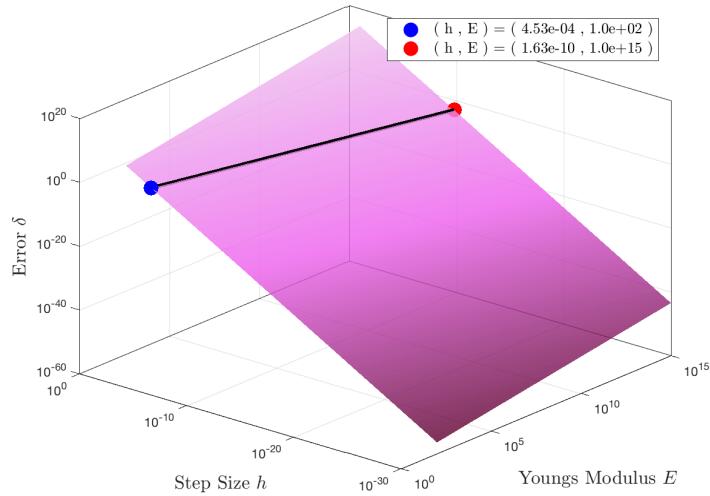
We saw in the previous experiment that there we would clearly not gain anything from using a second order finite difference approximation. On the other hand, there could have been something wrong, and the lines were alarmingly linear, thus doing the crossvalidation for a second order finite difference approximation for the stiffness tensor with one-sided forward Euler is presented

$$\varepsilon = \left\| h^2 \left(\frac{-\bar{k}(\bar{x} + 2\bar{v}) + 4\bar{k}(\bar{x} + \bar{v}) - 3\bar{k}(\bar{x})}{2\|\bar{v}\|_2} \right) - h^2 \mathbf{K} \bar{v} \right\|_\infty \quad (7.8)$$

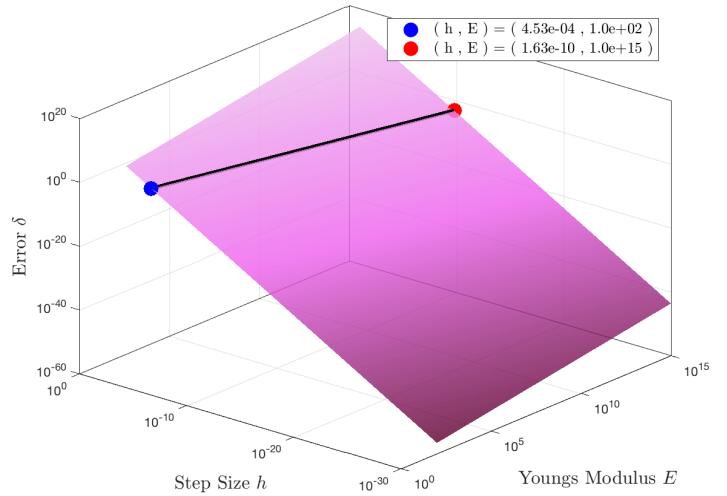
The crossvalidation for the parameters of Young's Modulus of which is describing the stiffness of the material and the time step size, is shown in figure 6 and 7. This is done by taking the absolute error for every value of h and E , which results in a 100×100 matrix of errors for all possible combinations of h and E .

This is showing the relation between the parameters. I am trying to connect this idea to a feasible area - an area of which we can choose the parameters as we wish and still achieve convergence. And see if the options of parameter selection gets better if we choose a different model.

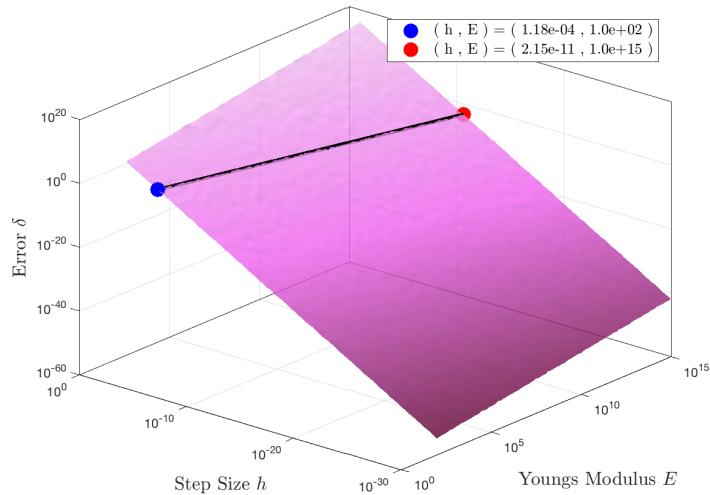
(a) Linear Elastic Material, \bar{e}



(b) St. Venant-Kirchhoff Material, \bar{e}



(c) Linear Elastic Material, \bar{r}



(d) St. Venant-Kirchhoff Material, \bar{r}

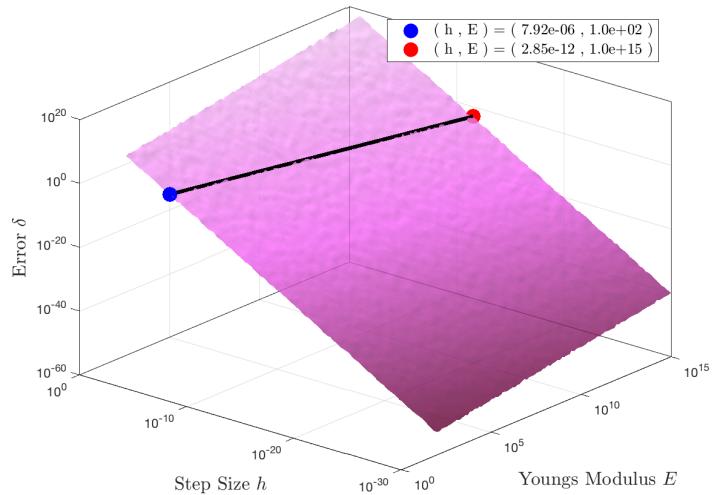


Figure 6: Crossvalidation of Young's Modulus, E , and Step Size h , with the absolute error, δ , depicted on the z-axis. The first row is a test case of 30 different unit vectors and the second row is a test case of 30 random vectors for every value of E and h on a 1st order approximation. The black line represents a feasible area for a tolerance equal to $\delta = 10^{-3}$. The blue point is the minimum value of h for the smallest value of E . The red point is the minimum value of h for the largest value of E to achieve an error no higher than the tolerance.

By a visual inspection of the figures in 6 it seems that there is not any diversity between the two models of Linear Elastic material and St. Venant-Kirchhoff material, when computing the absolute error with unit vectors, \bar{e} , especially based on the values of the points describing the lowest h we can get away with and still be in the error tolerance of 10^{-3} , for both the smallest and largest value of E . These values are the exact same for both models, which for smallest $E = 10^2$ is $h_{lin} = 4.53 \cdot 10^{-4} = h_{svk}$ and for the larges values of $E = 10^{15}$ is $h_{lin} = 1.63 \cdot 10^{-10} = h_{svk}$.

If we look how the two models differs by handeling random generated vectors, \bar{r} , one would easily jump to the conclusion of using the Linear Elastic material model, as it shows the lowest result of h , for the smallest value of $E = 10^2$, $h_{lin} = 1.18 \cdot 10^{-4} \neq 7.92 \cdot 10^{-6} = h_{svk}$ and for the larges value of $E = 10^{15}$ $h_{lin} = 2.15 \cdot 10^{-11} \neq 2.85 \cdot 10^{-12}$. It means that for the smallest value of Young's Modulus the two models differs with two orders in magitude, and one order in magnitude for the largest value of Young's Modulus. But the uncertainties that lie behind using random vectors, can also just make it a coincidence.

Thus thinking that using the mean and variance of simulated displacements, would be advantageous.

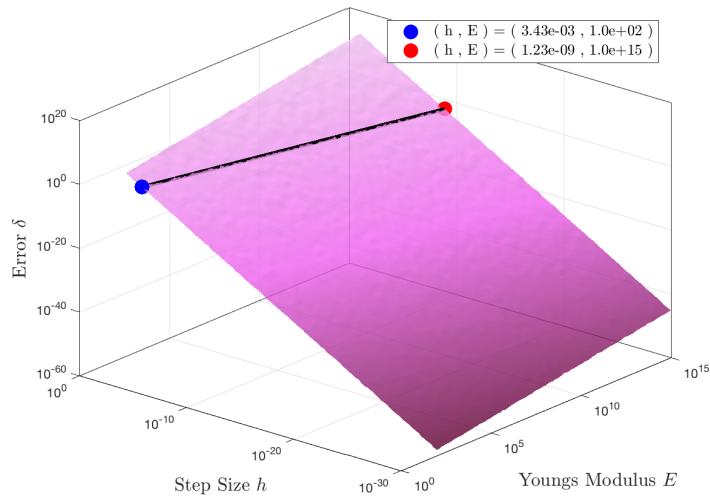
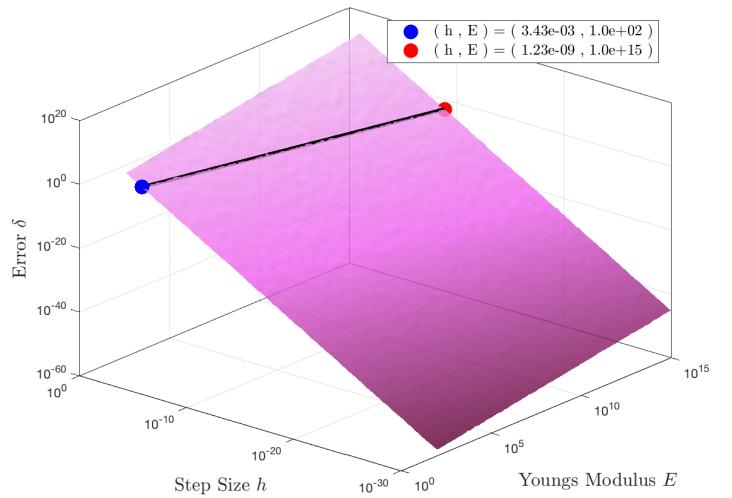
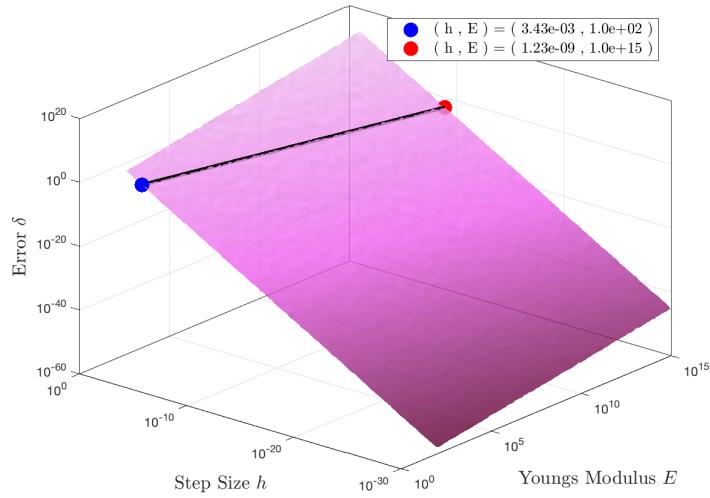
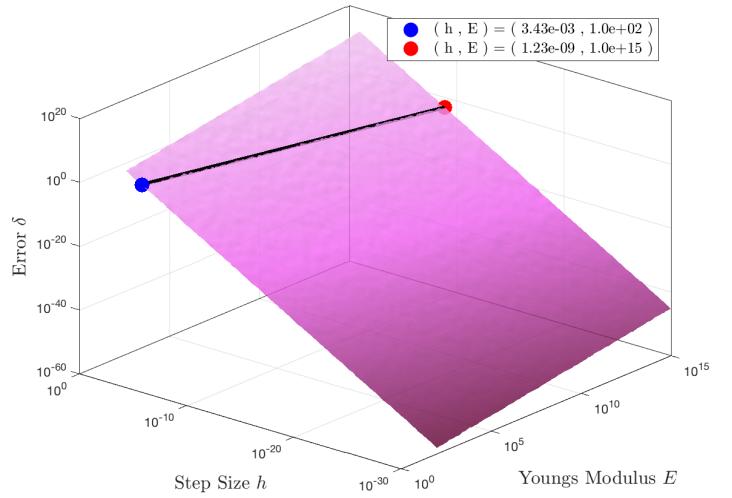
(a) Linear Elastic Material, $\bar{\iota}$ on 1st order

 (b) St. Venant-Kirchhoff Material, $\bar{\iota}$ on 1st order

 (c) Linear Elastic Material, $\bar{\iota}$ on 2nd order

 (d) St. Venant-Kirchhoff Material, $\bar{\iota}$ on 2nd order


Figure 7: Crossvalidation of Young's Modulus, E , and Step Size h , with the absolute error, δ , depicted on the z-axis. The first row is a test case on a first order approximation, whereas the second row is a test case on a second order approximation of the stiffness tensor. Both rows are tested against 30 random generation vectors, $\bar{\iota}$ in equation (7.3), with mean and variance from a real simulation. The black line represents a feasible area for a tolerance of $\delta = 10^{-3}$. The blue point is the minimum value of h for the smallest value of E . The red point is the minimum value of h for the largest value of E to achieve an error no higher than the tolerance.

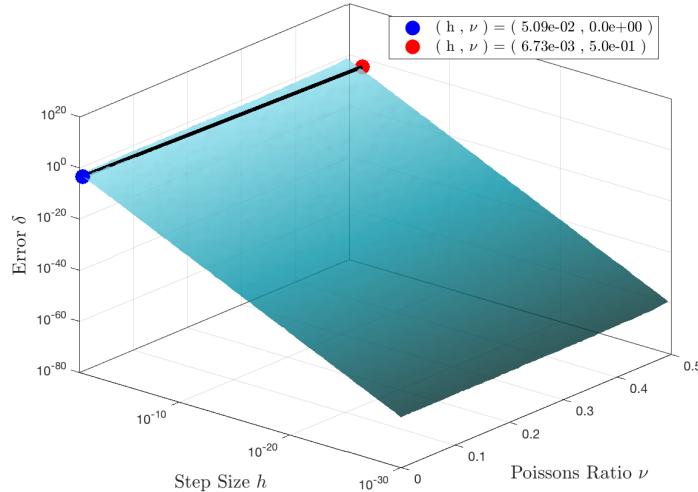
When looking at the figures in 7 of the crossvalidation of E and h done with the random generated vectors with mean and standard variance computed from simulated displacements, we see that there is absolutely no difference between the models or of the orders of finite difference approximations. This is also supported by the experiment in section 7.2, that it would give us no advantage using a 2nd order approximation. We see that the values of h is the exact same for all for sub figures $h_{lin1} = h_{lin2} = h_{svk1} = h_{svk2} = 3.43 \cdot 10^{-3}$ for the smallest value of $E = 10^2$ and for the largest value of $E = 10^{15}$ we have $h_{lin1} = h_{lin2} = h_{svk1} = h_{svk2} = 1.23 \cdot 10^{-9}$.

The areas that the black lines represent are not really some good feasible areas to be in, as we would actually like to be able to simulate stiff materials, in a physical range, and by the full implicit discretization being able to do this with a small value of h . This is also due to the tolerance of 10^{-3} set by me. A higher tolerance would naturally give us a larger feasible area, from which we can choose a smaller value of h , but would it be worth it if it costs us a an order of magnitude in error?

As shown in the figures 6 and 7, the dependence of E is very much existing, and this could also very much be the reason why we do not achieve convergence – maybe the approximation is simply not applicable to values of E in a real physical range? This is easy to overcome by using a non-physically based model, where the visual effect may only be presented as looking “real”.

We saw from the cross checking of E and h that best set of test vectors is within the random generated vectors with a mean and variance from simulated displacements on a first order finite difference approximation. Thus I have chosen only to display these settings for the crossvalidation of Poisson’s Ratio, ν , and step size, h , as they exhibit the same behaviour in change of model and test vectors. If the reader wishes to examine the other settings as in the crossvalidation of E and h , I refer the reader to appendix B.

(a) Linear Elastic Material, \bar{t} 1st order approximation



(b) St. Venant-Kirchhoff Material, \bar{t} 1st order approximation

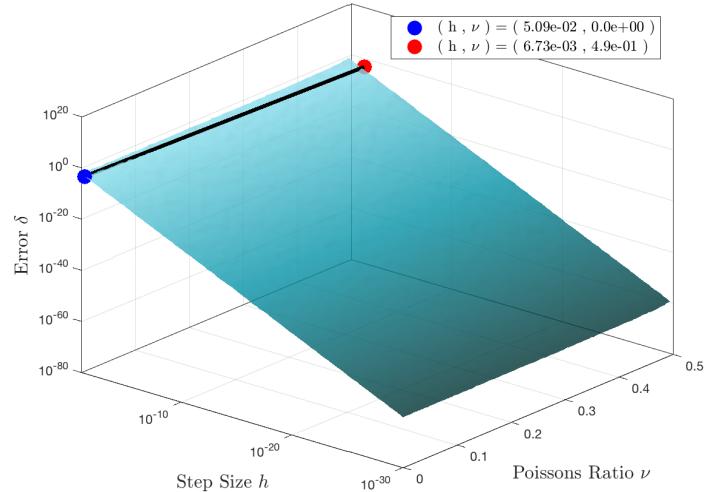


Figure 8: Crossvalidation of Poisson’s Ratio, ν , and Step Size h , with the absolute error, δ , depicted on the z-axis. It is a test case on a first order approximation of the stiffness tensor. The case is tested against 30 random generated vectors, \bar{t} in equation (7.3), with mean and variance from a real simulation. The black line represents a feasible area for a tolerance equal to $\delta = 10^{-3}$. The blue point is the minimum value of h for the smallest value of ν . The red point is the minimum value of h for the largest value of ν to achieve an error no higher than the tolerance.

Figure 8 is showing the result of crossvalidating 100 values og h in range $[10^{-2}, 10^{-15}]$ and 100 values of ν in the range $[0, \frac{1}{2}]$, with 30 test vectors \bar{t} from equation (7.3), for both Linear elastic material and St. Venant-Kirchhoff material.

The two figures 8a and 8b could indicate that there is no disparity between the two material models. This is highly indicated by looking at the values for the blue and the

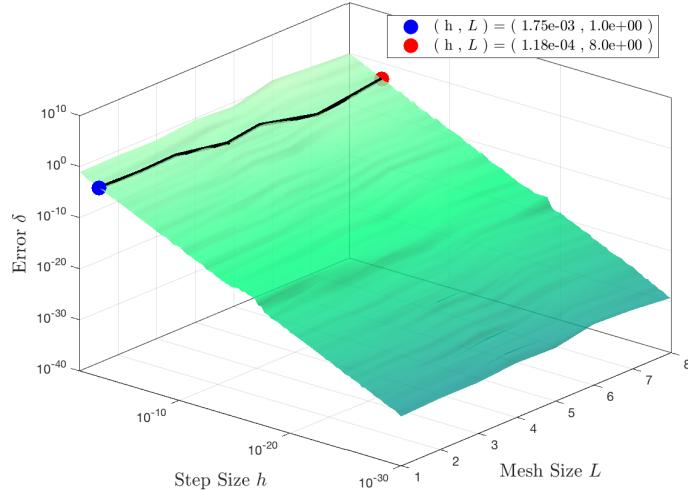
red point represents, which is identical for both models with $h_{lin} = 5.09 \cdot 10^{-2} = h_{svk}$ for the smallest value of Poisson's Ratio, $\nu = 0$, and for the largest value, $\nu = 0.49$, we have $h_{lin} = 6.73 \cdot 10^{-3} = h_{svk}$, for a tolerance of 10^{-3} .

Notice that the black line approximately follows the ν -axis. This can be interpreted as ν does not have an extensive significance, hence it can be chosen freely. Though we can see that the value of h from the smallest to the largest value of ν differs with an order of magnitude, which could mean that when trying to simulate a material with a relatively large value of Poisson's Ratio, such as rubber, the step size should be decreased with an order of magnitude. In addition, the step size, h , is what seems to have the bigger impact on the absolute error.

The next crossvalidation is of the mesh size, L and the step size h , where $h \in [10^{-2}, 10^{-15}]$ and $L = \{1, 2, 3, 4, 5, 6, 7, 8\}$ is the mesh number. The properties of the mesh number can be found in table 4.

As for the same reasons as the crossvalidation for ν and h , I will exclude the cases for test vectors of \bar{e} , \bar{r} and \bar{t} for the second order finite difference approximation. For these cases I refer the reader to appendix C.

(a) Linear Elastic Material, \bar{t} 1st order approximation



(b) St. Venant-Kirchhoff Material, \bar{t} 1st order approximation

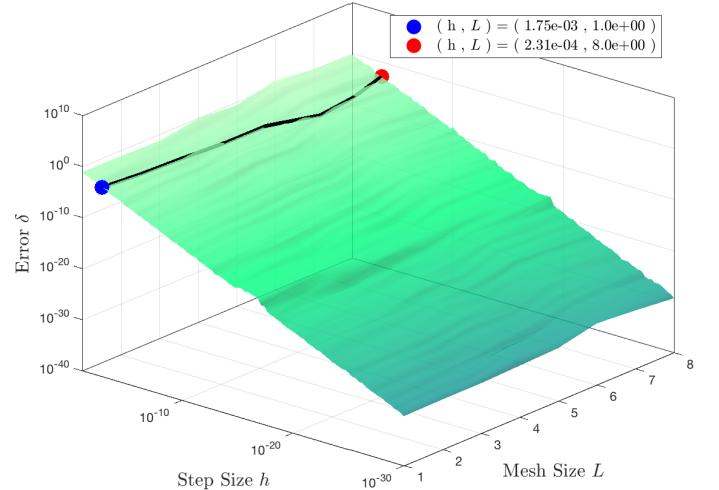


Figure 9: Crossvalidation of Mesh Size, L (referring to table 4 for info), and Step Size h , with the absolute error, δ , depicted on the z-axis. It is a test case on a first order approximation, which is tested against 30 random generation vectors, \bar{t} in equation (7.3), with mean and variance from a real simulation. The black line represents a feasible area for a tolerance equal to $\delta = 10^{-3}$. The blue point is the minimum value of h for the smallest mesh L_1 . The red point is the minimum value of h for the largest mesh L_8 to achieve an error no higher than the tolerance.

The behaviour of this crossvalidation of L and h , resembles the behaviour from the crossvalidation of ν and h . Where there is no distinction between the two material models in 9a and 9b.

When studying the black line that represents the feasible area, it can be seen that it is piecewise linear, thus it can be interpreted that for a given value of h , there is a mesh size there is more qualified in obtaining convergence inside the tolerance. Otherwise

there is no apparent difference between the models, with $h_{lin} = 1.75 \cdot 10^{-3} = h_{svk}$ for the first mesh L_1 and for the last mesh L_8 we have $h_{lin} = 1.18 \cdot 10^{-4} = h_{svk}$.

Table 4: The Properties of Mesh No. Used in Crossvalidation for both 2D og 3D.

Mesh No.	Vertices	Tetrahedra	$l_{min} [m]$
L_1	300×3	1098×4	0.80
L_2	325×3	1197×4	0.75
L_3	576×3	2293×4	0.65
L_4	1701×3	7737×4	0.50
L_5	1782×3	8150×4	0.45
L_6	2500×3	11741×4	0.40
L_7	4173×3	20423×4	0.35
L_8	5577×3	27977×4	0.30

7.4 2D Experiments

The experiments and examples in this section is exclusively based on the previous section of crossvalidation of the parameters. The goal is to compare the parameter selection on a 2D scale as it can be hard to interpret the difference en both model selection, order of finite difference approximation and contrasting vectors, given a 3D plane.

In table 5 is presented all various options for material models, order of approximation and vectors, given in a color coding used for the 2D figures in this section.

Table 5: Table of Correspondance Between Colors, Models and Test Vectors for viewing the Crossvalidation in 2D in figures 10, 11, 12 and 13.

lin _e 1st	Linear Elastic Material with unit vectors on 1st order approximation
lin _r 1st	Linear Elastic Material with random vectors on 1st order approximation
lin _{irl} 1st	Linear Elastic Material with "real" vectors on 1st order approximation
lin _{irl} 2nd	Linear Elastic Material with "real" vectors on 2nd order approximation
svk _e 1st	St. Venant-Kirchhoff Material with unit vectors on 1st order approximation
svk _r 1st	St. Venant-Kirchhoff Material with random vectors on 1st order approximation
svk _{irl} 1st	St. Venant-Kirchhoff Material with "real" vectors on 1st order approximation
svk _{irl} 2nd	St. Venant-Kirchhoff Material with "real" vectors on 2nd order approximation

As seen from the 3D figures from the crossvalidation of Young's Modulus, E and step size h , we should be able to achieve some results within the tolerance of 10^{-3} with a value of $h = 10^{-4}$. Thus I have chose to present a 2D example of this scenario, and find a feasible area thereof.

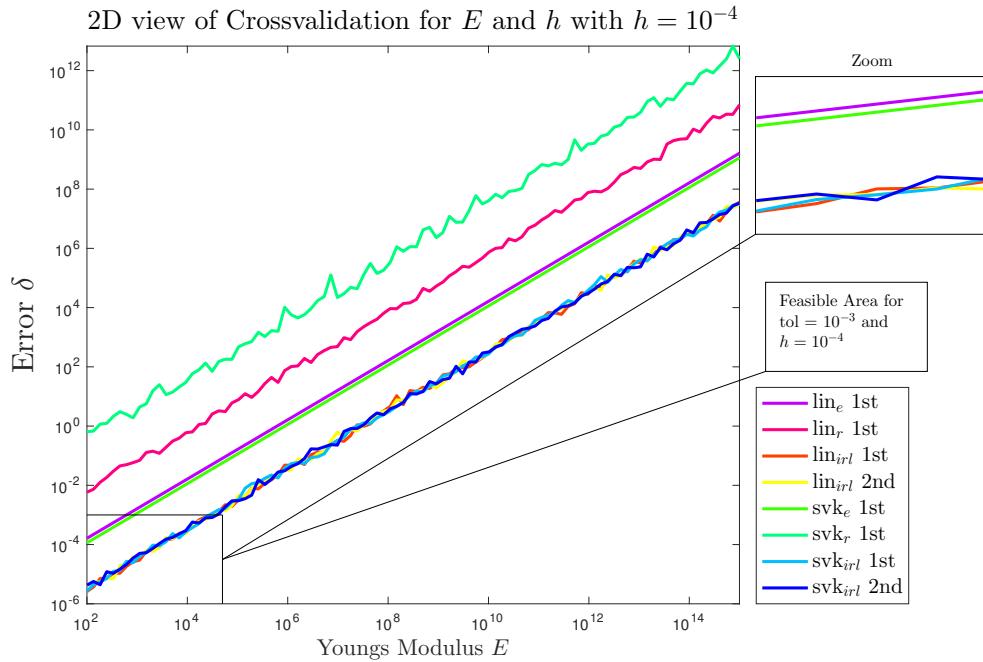


Figure 10: 2D view of Crossvalidation of E and h , depicted with a fixed value of $h = 10^{-4}$ and various values of E . The outlined box in the lower left corner illustrates a feasible area for which one can choose a value of E and still comply with the tolerance of $\delta = 10^{-3}$. The figure in the upper right corner is a zoomed version of the feasible area.

From figure 10 it can be seen that with this selection of parameters, we cannot achieve a convergence area for all combinations. If we wanted our feasible area to include all cases in table 5, we would either have to decrease the value of h or increase the acceptable tolerance, where neither is desirable. We also see a confirmation on that there is no advantage in using a second order approximation. This can be seen by looking at the zoomed area in figure 10, where four lines are tangled together, these lines are for test vectors \bar{t} , both material models and both first and second order finite difference approximations.

Notice the range of E in the feasible area in figure 10. This range is approximately limited to values of $E \in [10^2, 6 \cdot 10^4]$. This range does not even include the elasticity of foam, which according to [15] is approximately $2.5 \cdot 10^6 - 7 \cdot 10^6$ and is the material with the lowest Young's Modulus reported on the website. It could turn out to be nearly impossible to make use of the approximation of the stiffness tensor for a real physical range of Young's Modulus, which extends all the way up to values of approximately 10^{15} .

In the 3D figures in 8a and 8b from the crossvalidation of ν and h , we saw that the value of $h \approx 10^{-2}$ were respectable when considering the tolerance of 10^{-3} . Thus this is what fixed value I chose to depict in 2D. Instead of only looking at the combinations in figure 8, all combinations in table 5 are included.

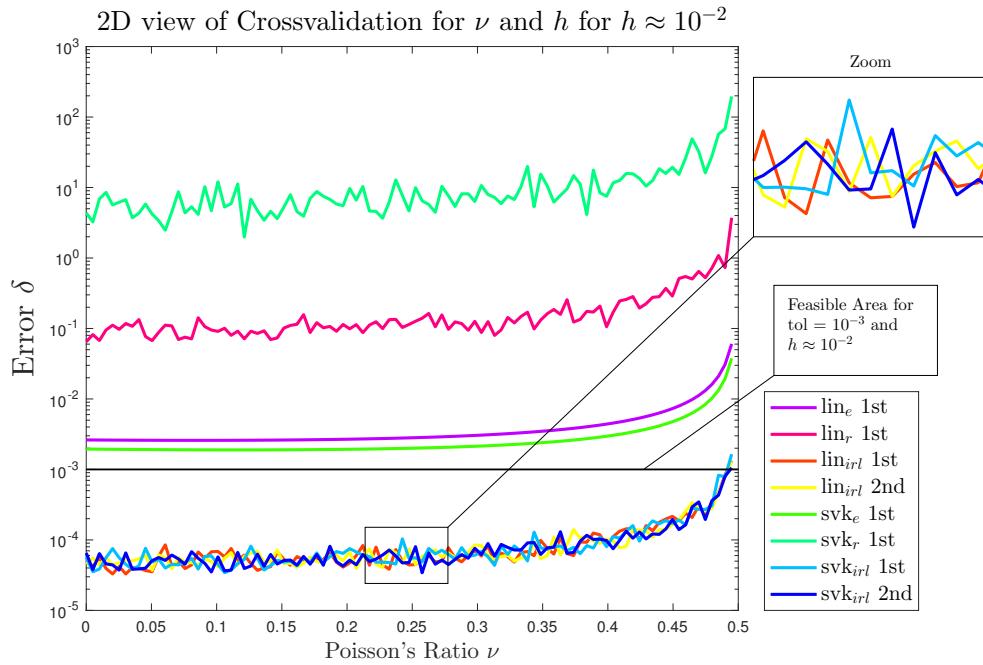


Figure 11: 2D view of Crossvalidation for ν and h for a fixed value of $h \approx 10^{-2}$. The box in the upper right corner is a zoom in of a part of what lies within the feasible area of parameter selection for a tolerance of 10^{-3}

When looking at the crossvalidation of ν and h in 2D it is highly visible how insignificant the value of ν is. Figure 11 imply that there is a diversity between the material models when the choice of vectors is random, \bar{r} , or unit vectors, \bar{e} .

The feasible area for the tolerance of 10^{-3} , includes both material models, but only for random generated vectors, \bar{r} . We see that we can freely choose a value of ν along its entire range, and still achieve convergence, within the tolerance.

In figures 9a and 9b, from the crossvalidation of mesh size, L and step size, h , we saw that there was no considerable diversity in the choice of mesh size. We also found that we should expect results when choosing $h = 10^{-4}$. Hence, the 2D example is for a fixed value of $h = 10^{-4}$. This is depicted in the figure below for all combination shown in table 5. For properties of the mesh numbers look at table 4.

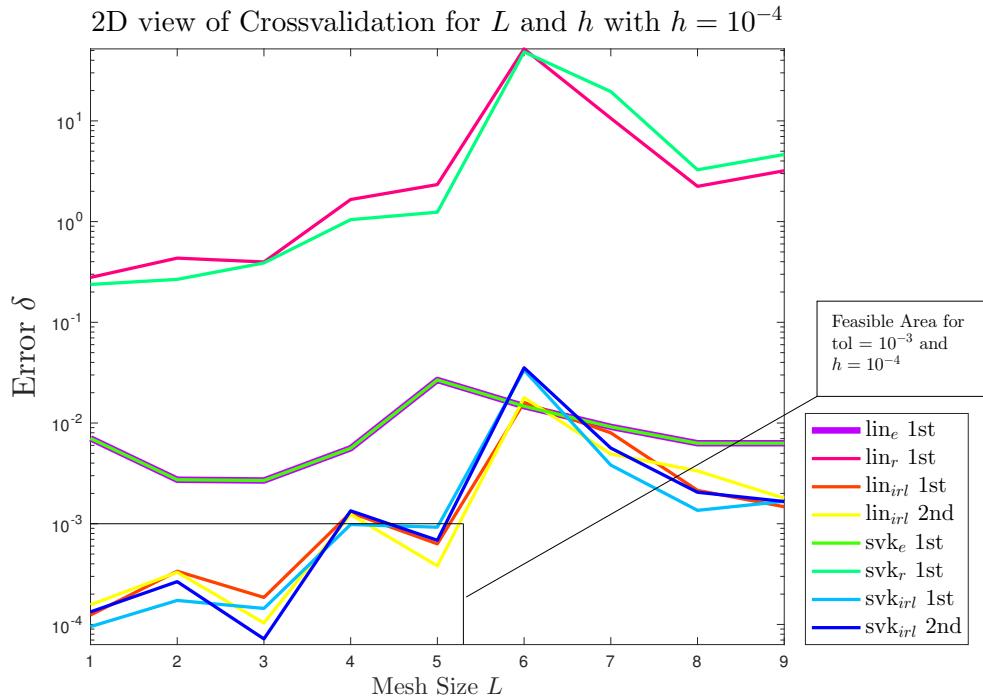


Figure 12: 2D view of Crossvalidation of L and h , depicted with a fixed value of $h = 10^{-4}$. In the lower left corner is outlined a box, which describes a feasible area for at tolerance equal to 10^{-3} .

From the appearance of figure 12 in two dimensions, it is insinuated that there is a considerable diversity in choosing the size of the mesh, which is not what I expected from the experiment in three dimensions. For some unknown reason mesh number 6 is unexplainably higher than both smaller or larger meshes. This peak of at least an order in magnitude is consistent for random vectors, \bar{r} and \bar{t} .

When only considering the feasible area constructed from a tolerance of 10^{-3} , we see that it includes both material models, but only by the use of random vectors with mean and variance from real simulated displacements. By a visual inspection seems reasonable to assume that mesh number 3, is the best mesh to use, in order to obtain a small error.

Maybe we can minimize the value of h , by choosing a fixed value of the mesh size to be L_3 . After that we assumed that the most profitable mesh were L_3 , I fixed the value of L to 3 and we can now analyse what range h should be within, where we still comply with the tolerance of 10^{-3} .

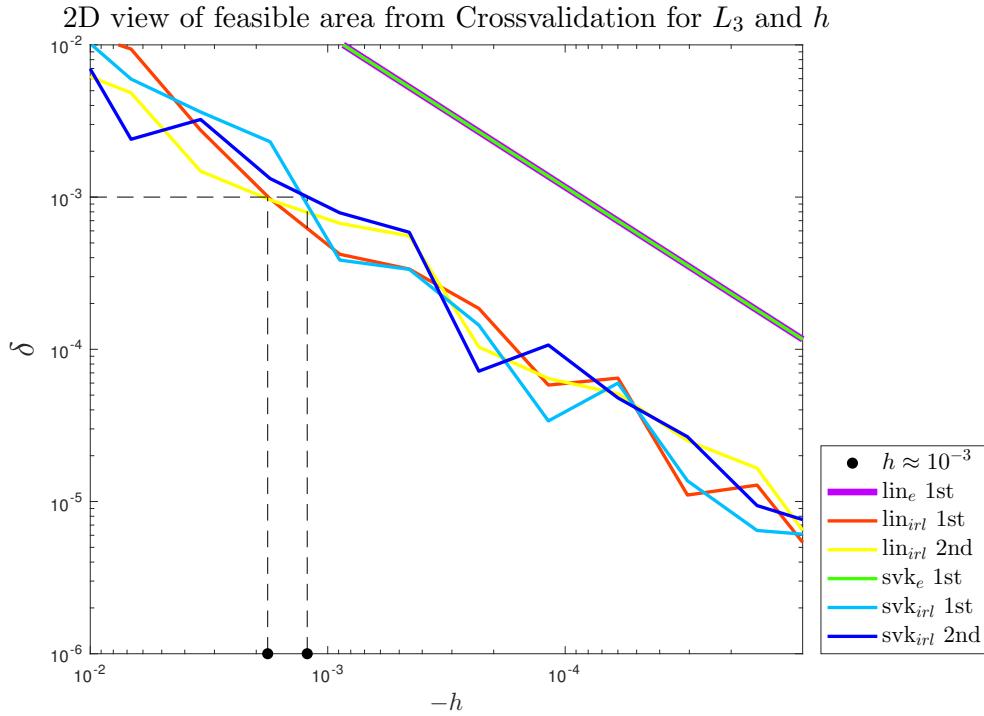


Figure 13: 2D view of the Crossvalidation of L and h for a fixed value of $L = L_3$ meaning mesh no. 3 described in table 4. The points portray the least value of h within the tolerance of 10^{-3} .

By fixing L to a value of 3, we are able to increase the step size a full order in magnitude and still be within the tolerance, which without a doubt is beneficial. The increase in h , is depicted in figure 12 by the black points on the h -axis.

8 Conclusion

During the last section there have been presented a great deal of experiments, namely in order to reach a conclusion of why the finite difference approximation is not living up to the expectations.

We found out that to a limited values of h and random generated values with mean and variance from simulated data, the matrix \mathbf{A} is both symmetric and positive definite – meaning that the properties needed for using preconditioned conjugate gradient is not flawed by the use of a first order finite difference approximation.

Another concern was that the order of the approximation. Is a first order approximation enough? The answer found is a definite yes. Though, we might try reasoning about why the elastic forces as function of τ , is perfectly straight, with absolutely no outliers. Even if I had a sample of 10, they were still dead straight. But the conclusion to this may simply be, that the elastic forces are without a doubt just linear, which in the end leads us to believe that there is no advantage in using a second order finite difference approximation.

From the crossvalidation we tried to obtain a feasible area for Young's Modulus, E , Poisson's Ratio, ν , mesh size, L , and step size, h . The cross checking of E and h ,

indicated that the range from which we could choose the step size depended highly on the value of Young's Modulus. E.g. if simulating a body with a large stiffness with a value of $E \approx 10^{15}$ we would need a step size of at least $h = 10^{-9}$ to be within the tolerance of 10^{-3} . On the other hand, if we look at a more reasonable step size $h = 10^{-4}$, the range of E is not in a physical range. This supports why the approximation provides results when applied to a non-physically based model in [11].

When considering the importance of E , the impact of ν and L subsides. Although we found out that some meshes produce a slightly smaller error than others. I do not think that this has a great impact, when examining the whole picture.

There are indications of that a numerical approximation is not applicable to a physical-based model, as the feasible area for E , does not contain real physical values.

When everything is taking into account, it seems we are not gaining any advantage over the exact version of the stiffness tensor. As it looks now at least.

9 Further Work

As we did not get any results from the actual simulation, it would be a place to start – see if we could actually simulate with Young's Modulus, E for unrealistic soft object, and get some reasonable results. The results of the experiments is suggesting that this is achievable.

If time were not an everlasting enemy, I would double check that the error made, were not caused by me, my understanding, or my implementation.

It is also conceivable that the reason for not achieving any simulation results, does not even lie within the aspects investigated during this paper. A new focus could be to investigate the boundaries for fixed object and applicable forces for free-floating objects.

If the boundaries is attached to the stiffness tensor, then it would make perfect sense, as we remove the stiffness tensor and substitute in a first order approximation formed by the elastic forces.

If we had a case where the elastic forces were linear, an adaptive time stepping method, could cause a more a smaller error of the numerical approximation.

References

- [1] BONET, J., AND WOOD, R. D. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge university press, 1997.
- [2] CHENTANES, N. Deformable body simulation on gpu, April 2008. <http://developer.download.nvidia.com/SDK/10/direct3d/Source/DeformableBody/doc/DeformableBody.pdf>.
- [3] ERLEBEN, K. Source code: Hyper sim, November 2011. <https://github.com/erleben/hyper-sim>.
- [4] ERLEBEN, K. *Finite Element Modeling of Hyper Elastic Materials*. Unpublished, 2013.
- [5] GONZALEZ, O., AND STUART, A. M. *A First Course in Continuum Mechanics*. Cambridge University Press, 2008.
- [6] KINCAID, D. R., AND CHENEY, E. W. *Numerical analysis: mathematics of scientific computing*, vol. 2. American Mathematical Soc., 2002.
- [7] LAI, W. M., RUBIN, D. H., RUBIN, D., AND KREML, E. *Introduction to Continuum Mechanics*. Butterworth-Heinemann, 2009.
- [8] LARSON, M. G., AND BENGZON, F. *The Finite Element Method: Theory, Implementation, and Applications*. Springer, 2013.
- [9] MCADAMS, A., ZHU, Y., SELLE, A., EMPEY, M., TAMSTORF, R., TERAN, J., AND SIFAKIS, E. Efficient elasticity for character skinning with contact and collisions. In *ACM Transactions on Graphics (TOG)* (2011), vol. 30, ACM, p. 37.
- [10] SAYAS, F.-J. A gentle introduction to the finite element method. *Cited on* (2008), 40.
- [11] SIFAKIS, E., AND BARBIC, J. Fem simulation of 3d deformable solids: A practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses* (2012), ACM, p. 20.
- [12] SIN, F. S., SCHROEDER, D., AND BARBIĆ, J. Vega: Non-linear fem deformable object simulator. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 36–48.
- [13] TAYLOR, J. R. Error analysis: The study of uncertainties in physical measurements. *Sausalito: University Science Book* (1997).
- [14] Infinitesimal strain theory, July 2016. https://en.wikipedia.org/wiki/Infinitesimal_strain_theory.
- [15] Young's modulus, August 2016. https://en.wikipedia.org/wiki/Young%27s_modulus.
- [16] WALCZAK, Z. Graphics in latex using tikz. *EuroBachotEX 2007* (2007), 176.

Appendices

A Symbolic Differentiation

Table 6: Symbolic Differentiation of The Material Elasticity Tensor in MATLAB

```
1 % Material Elasticity Tensor
2 A = sym('TEMP')*ones(9,9);
3 for i=1:3
4     for j=1:3
5         for k=1:3
6             for m=1:3
7                 r = (i-1)*3 + k;
8                 c = (j-1)*3 + m;
9                 A( r, c ) = diff( P(i,j) , F(k,m));
10            end
11        end
12    end
13 end
14 A = simplify(A)
15 matlabFunction(A, 'file', 'compute_A.m');
```

B Crossvalidation: $\nu \times h$

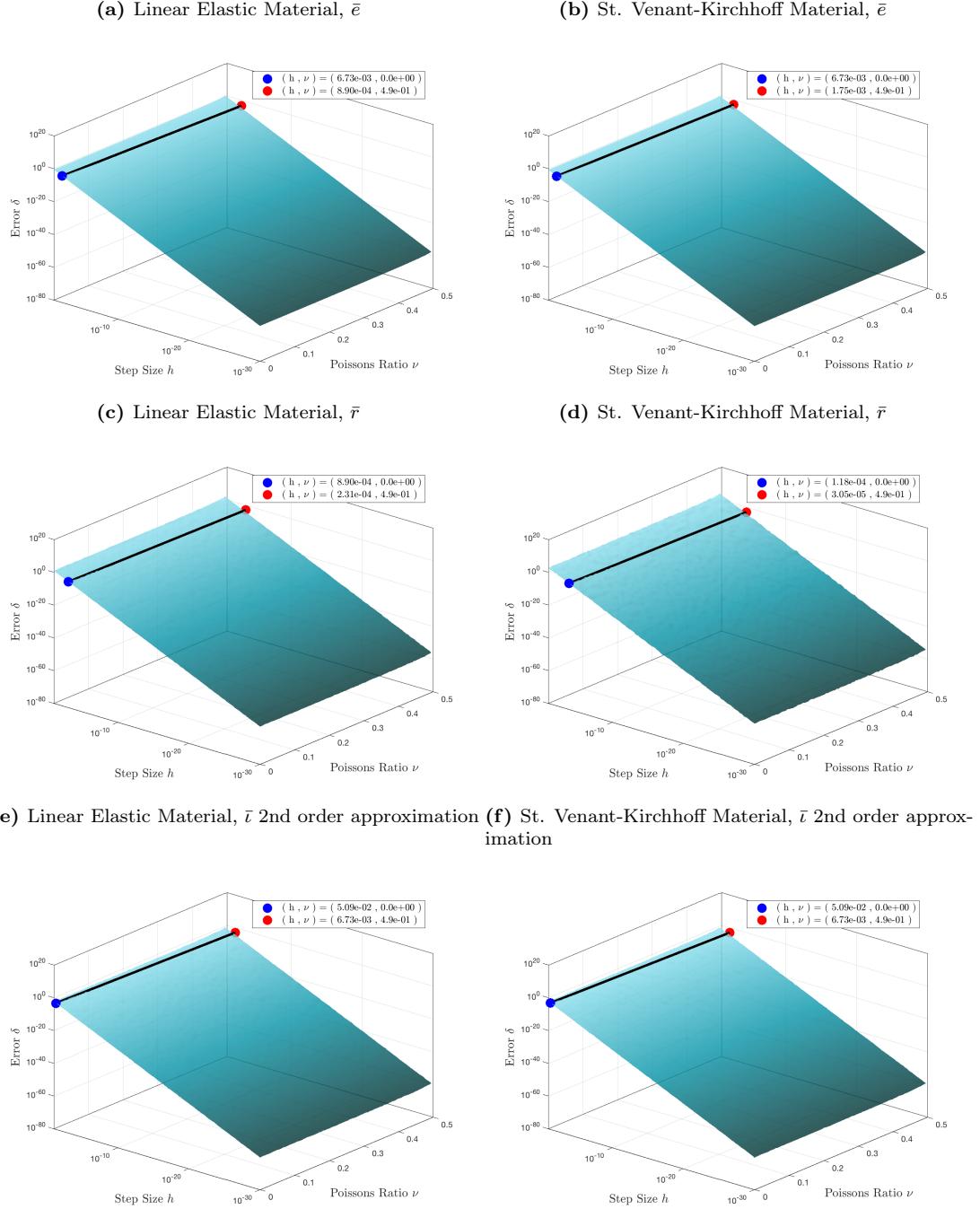


Figure 14: Crossvalidation of Poisson's Ratio, ν , and Step Size h , with the absolute error, δ , depicted on the z-axis. The first row is a test case of 30 different unit vectors and the second row is a test case of 30 random vectors for every value of E and h on a 1st order approximation. The third row is a test case of 30 random generated vectors on a second order approximation of the stiffness tensor. The black line represents a feasible area for a tolerance equal to $\delta = 10^{-3}$. The blue point is the minimum value of h for the smallest value of ν . The red point is the minimum value of h for the largest value of ν to achieve an error no higher than the tolerance.

C Crossvalidation: $L \times h$

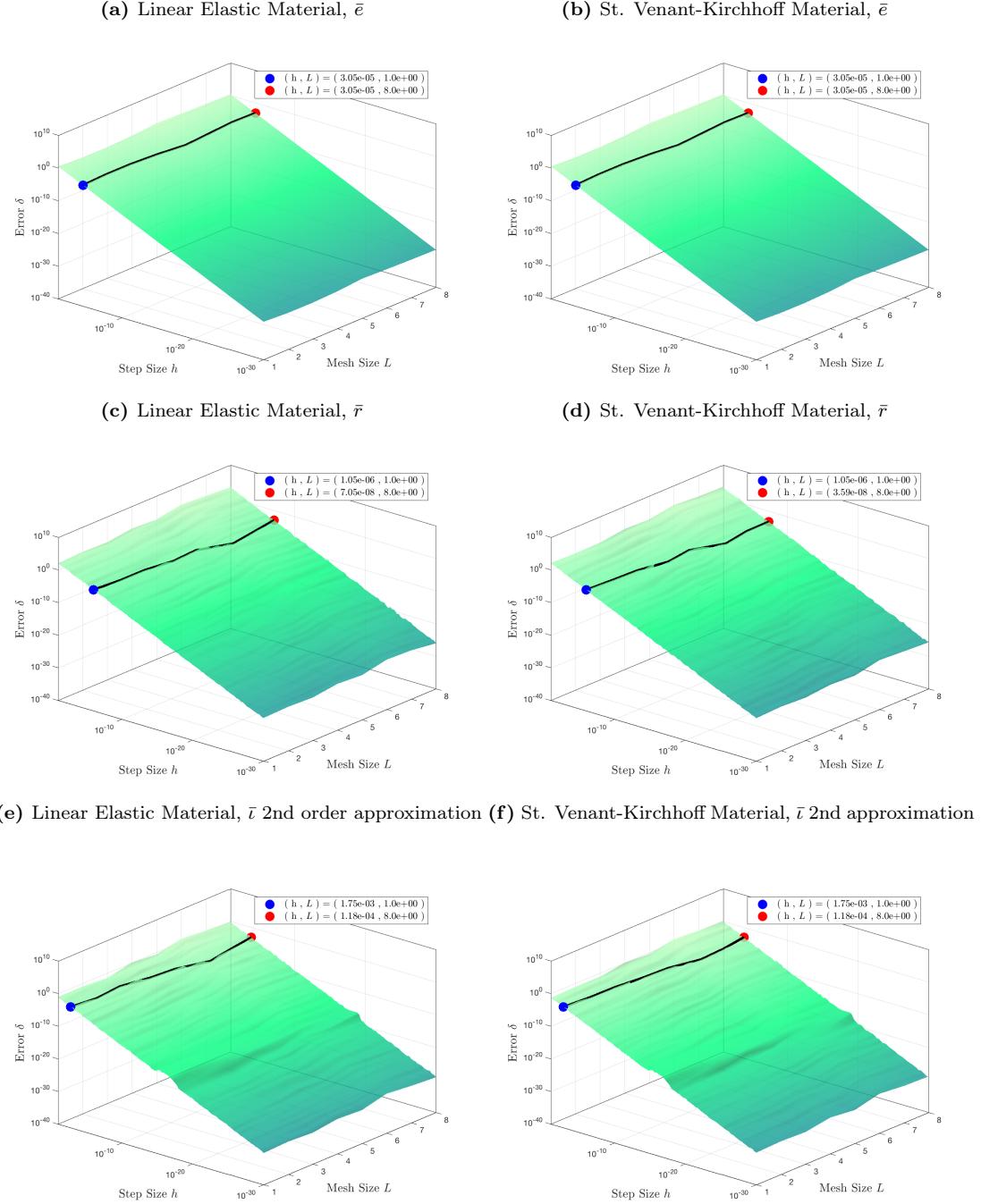


Figure 15: Crossvalidation of Mesh Size, L (referring to table for info 4), and Step Size h , with the absolute error, δ , depicted on the z-axis. The first row is a test case of 30 different unit vectors and the second row is a test case of 30 random vectors for every value of E and h on a 1st order approximation. The third row is a test case of 30 random generated vectors on a second order approximation of the stiffness tensor. The black line represents a feasible area for a tolerance equal to $\delta = 10^{-3}$. The blue point is the minimum value of h for the smallest mesh, L_1 . The red point is the minimum value of h for the largest mesh, L_8 , to achieve an error no higher than the tolerance.