A short introduction to soft tissue simulation

Stefan Suwelack (ed.)

suwelack@kit.edu

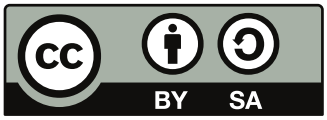
June 15, 2015

Please help to make this book better by reporting errors or any other issue at
https://github.com/ssuwelack/soft-tissue-simulation/issues.

Are you a researcher? Please add a reference to your soft tissue simulation related work in the application chapter and submit a pull request.

This is a joint work in progress. If you feel you have something to contribute to an introductory text for soft tissue simulation, please become an author of this book!

Editor:
Stefan Suwelack, suwelack@kit.edu



Contents

Part I.

Theory

The skill of writing is to create a context in which other people can think.

— Edwin Schlossberg

1.

Introduction

Modeling soft tissue by means of continuum mechanics has become an active area of research in many fields such as patient-specific surgery simulation, non-rigid image registration and cardiovascular diagnostics. However, it currently is a very young discipline that evolves at a rapid pace (and has arguably to mature much more before these methods become part of the daily clinical routine). When I was looking for literature that could be recommended to my students I felt that there is currently no real introductory textbook on this matter. This little script is to serve exactly this purpose. Although it is in most parts a short course on elasticity theory and the finite element method (FEM), it should also hint to recent research results in terms of material laws, special FEM algorithms and applications of soft tissue simulation.
The intended audience of this text are postgraduate level students with a computer science or engineering background. Its purpose is to introduce the most important ideas, principles and methods in the area of soft tissue simulation and it should serve as an introductory text either for self study or accompanying a lecture. It was in particular written with two specific goals in mind. On the one hand you should be able to implement your own simple soft tissue simulation using a linear FEM algorithm once you've completed this course. On the other hand the script should give you a general idea on the topic and should serve as a good starting point for further studies.
For the impatient, here is a list of the topics currently covered by this text:

- Introduction to elasticity: Strain, stress, balance principles and mechanical energy
- Material laws: Work conjugacy, linear elasticity, hyperelasticity, viscoelasticity
- The road to discretization: Variational formulation and weak form
- The finite element method: Shape functions, matrix formulation, numerical integration
- Time integration and projective constraint handling
- Efficient real-time models: Corotated tetrahedra
- Linear solvers: Conjugate gradients, preconditioners and multigrid

There are many great text books and internet resources about elasticity and the FE method. For further reading and in order to complement this text I recommend the following ones which I found exceptionally useful:

- The textbook *Nonlinear solid mechanics* by Gerhard Holzapfel provides thorough, yet concrete treatment of non-linear elasticity. It's a great read especially for engineers. The notation used in this text closely follows the notation by Holzapfel, so his book would make the perfect follow up to this script if you would like to really understand elasticity theory.
- If you prefer a highly mathematical treatment of the subject then the book *Mathematical foundations of elasticity* by Jerrold Marsden and Thomas Hughes is definitely worth a look.
- In my opinion the book *Nonlinear Finite Elements for Continua and Structures* by Belitschko et al. provides an excellent introduction to the finite element method and strikes a good balance between mathematical rigor and practical examples.

- If you would like to take a deeper look at error estimation methods as well as linear solver technology the book *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics* by Dietrich Braess is a useful read.
- For a more general introduction into numerics I highly recommend the book *Numerical Analysis* by Endre Süli and David MAyers
- In my opinion you cannot completely understand numerical methods without actually running them. That's why I highly recommend `www.solidmechanics.org` run by A.F. Bower where you can find lots of MATLAB sample code.

Although a detailed treatment of recent research results and applications in the context of soft tissue simulation is out of scope for this document, such work is referenced in the application chapter. If you know of any work that should be listed there, please submit a pull request or write me an e-mail.
I have always been fascinated how simulations turn equations and numbers into something that mimics reality. I am therefore convinced that learning about simulation methods must include seeing this process in action. In order to make this possible, this document contains a tutorial section which contains hands-on examples and exercises based on state-of-the-art open source simulation tools.
 My biggest wish is that it will be copied, edited, corrected and enhanced by anyone who finds it a good seedpoint for his/her own work. If you feel there is something missing (which there definitely is), please become an author of this book!
Finally I'd be delighted to hear from you if you found this text useful for your studies, thesis work or research. You can reach me by e-mail under suwelack(at)kit.edu.

Karlsruhe, Germany, 2015

Stefan Suwelack

If people do not believe that mathematics is simple, it is only because they do not realize how complicated life is.
— John von Neumann

2.

Biomechanical modeling of soft tissue

In this chapter the foundations of continuum mechanics based soft tissue modeling are introduced. We start by outlining the fundamentals of elasticity theory before presenting typical material models for biological soft tissue.

2.1. Getting started

Mechanics is often defined as *... a branch of physics concerned with the behaviour of physical bodies when subjected to forces or displacements, and the subsequent effect of the bodies on their environment*. You will most certainly remember some experiments from your physics class in school where you tried to calculate the paths of colliding billiard balls by using physical principles such as the energy and impulse conservation laws. If you have some background in thermodynamics you might be familiar with systems that consist of huge numbers of particles (statistical mechanics). If you had to deal with semiconductors during your studies you certainly have encountered the strange behavior of particles at the atomic and subatomic scales (quantum mechanics). Now, what is meant by the term *continuum mechanics* and why is this set of methods useful for soft tissue simulation?
Imagine you would like to design a computer-based training tool that allows a surgeon to practise the removal of a liver tumor (partial liver resection) before actually performing the surgery on a patient. During surgery the liver is not only subject to the forces created by surgical instruments, but also to the forces originating from breathing motions. In order to simulate the deformations (i.e. the displacement of all points inside the liver) that occur from these forces we first have to think of a suitable computer model. Keep in mind that we always want to solve the problem as fast as possible (in this case even in real-time). Thus, the model has to be as accurate as required, but as simple as possible. We would be very ill-advised to try to model the behaviour of each atom in the liver by quantum mechanics. Even the modeling of every cell in the liver would require computing power that is far beyond the capability of current hardware. To effectively model the mechanical behavior of the liver we have to turn to a macroscopic model. In this sense a liver consists of different materials such as healthy liver tissue, blood vessels or tumor tissue. We now assign these materials to the corresponding sections of the liver. Within each section the liver is assumed to be continuous. If we chop down a piece of the liver until we have an infinitesimal small element we will therefor not find any cells, molecules or atoms, but instead just an infinitesimal element of the selected material. We will later see that the analysis of infinitesimal elements is in fact what continuum mechanics is all about.

2.2. A short introduction to elasticity

2.2.1. Kinematics

We consider a body $\mathcal{B}$ that can be viewed as a continuous distribution of matter in space and choose a standard right-handed orthonormal coordinate system as the reference frame (Fig. 2.1). The body moves in space from one instant of time to another, occupying different geometrical regions $\Omega_0, ..., \Omega$ in the process. These regions are called configurations of $\mathcal{B}$ at time t. The configuration Ω_0 at time $t = 0$ is called the initial configuration while the configuration Ω at t is called the current configuration. Our goal

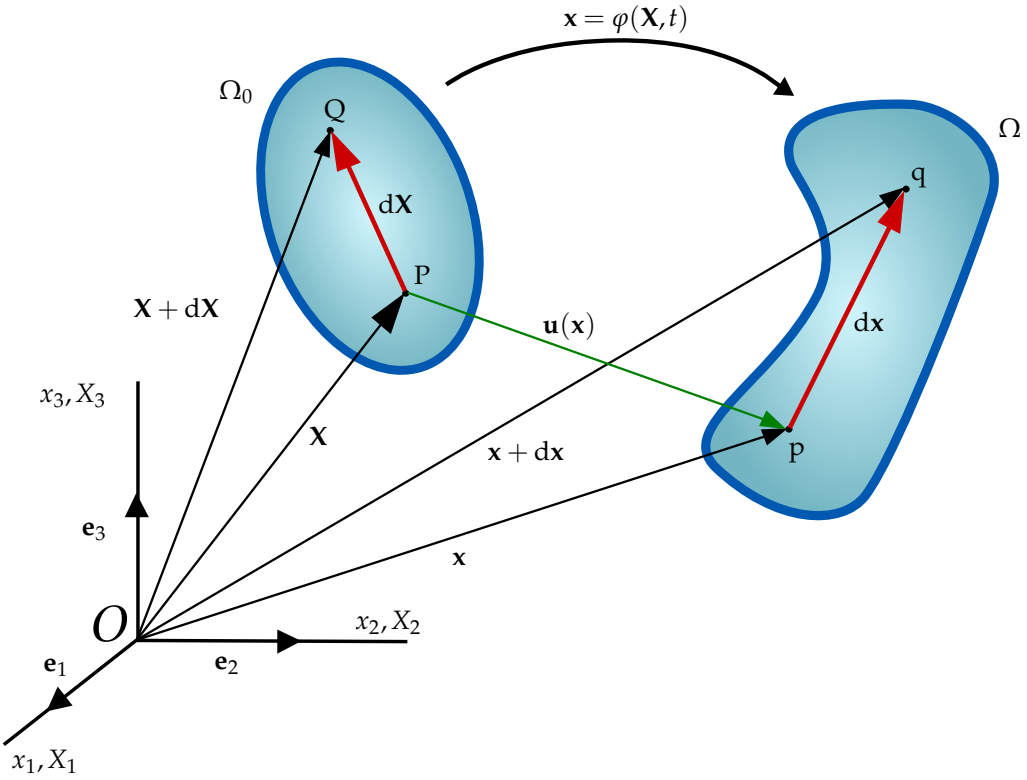


Figure 2.1.: Deformation of the body $\mathcal{B}$ with material points P, Q and infinitesimal line element d$\mathbf{X}$ from the reference configuration Ω_0 to the current configuration Ω (based on [Wik13])

is to describe the deformation of $\mathcal{B}$ with respect to a reference configuration. Throughout this text, we will always assume the initial configuration to be our reference configuration.
We start by analyzing a continuum particle $\mathbf{P} \in \mathcal{B}$. It is important to note that $\mathbf{P}$ has no point mass (as opposed to a discrete particle in Newtonian mechanics). The particle is at position $\mathbf{X}$ in the reference configuration and moves to the position $\mathbf{x}$ in the current configuration (Fig. 2.1). We define the vector field

$$\mathbf{x} = \varphi(\mathbf{X}, t) \tag{2.1}$$

that maps the positions $\mathbf{X} \in \mathcal{B}$ of all points in the reference configuration to their respective positions $\mathbf{x}$ in the current configuration. In our analysis we assume that φ possess a continuous derivative and that it is uniquely invertible, i.e. the inverse mapping

$$\mathbf{X} = \varphi^{-1}(\mathbf{x}, t) \tag{2.2}$$

exists. Here, some important terminology should be introduced. φ is also called the motion of the body $\mathcal{B}$ over time. If we look at φ at a specific point in time, φ is called the deformation of the body. Thus, we speak of deformation if we mean the motion of a body that is independent of time. A body undergoing a deformation can change its shape, position and orientation. If the deformation is constant for all $\mathbf{X} \in \mathcal{B}$, the deformation consists only of translations and rotations and is called a rigid-body motion. Please note that in contrast to the common definition of deformation which implies changes to the body's shape, the continuum mechanics definition also includes rigid body motions.
Often the deformation is described in terms of the displacement

$$\mathbf{u}(\mathbf{X}, t) = \mathbf{x}(\mathbf{X}, t) - \mathbf{X} \tag{2.3}$$

of the body $\mathcal{B}$.

Continuum mechanics can be regarded as describing the behavior of infinitesimal line, area and volume elements during the passage from the reference to the current configuration. The second order tensor

$$\mathbf{F}(\mathbf{X},t) = \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\mathbf{X}} = \frac{\mathrm{d}x_i}{\mathrm{d}X_j} = x_{i,j} = \nabla\mathbf{x} = \mathrm{Grad}\mathbf{x} \tag{2.4}$$

describes the relation between the spatial line element $\mathrm{d}\mathbf{x}$ and the material line element $\mathrm{d}\mathbf{X}$ and is called the deformation gradient. Here, we showcased several different notations for the deformation gradient: The vectorial notation, the index notation and its shortened version as well as the notation using the nabla and the material gradient (Grad) operator. For more information on the notation of variables and operators used in this thesis please refer to the glossary. Using the relation (2.3), the deformation gradient can also be expressed in terms of the displacement field $\mathbf{u}(\mathbf{x},t)$:

$$\mathbf{F}(\mathbf{X},t) = \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{X}} + \mathbf{I} = u_{i,j} + \delta_{ij} = \nabla\mathbf{u} + \mathbf{I} = \mathrm{Grad}\mathbf{u} + \mathbf{I} \tag{2.5}$$

Assuming that the derivative of the inverse mapping φ^{-1} exist, we can define the inverse of the deformation gradient

$$\mathbf{F}^{-1} = \frac{\mathrm{d}\mathbf{X}}{\mathrm{d}\mathbf{x}} = \mathrm{grad}\mathbf{X} \tag{2.6}$$

in an analogous manner. Please notice that the lowercase grad operator denotes the derivative with respect to $\mathbf{x}$ (spatial gradient).

2.2.2. Deformation of infinitesimal elements

In order to describe the change of infinitesimal volume elements due to deformation, we regard the parallelepiped that is spanned by the three non-coplanar line elements $\mathrm{d}\mathbf{X}^{(1)}, \mathrm{d}\mathbf{X}^{(2)}, \mathrm{d}\mathbf{X}^{(3)}$ at the point $\mathbf{X}$ in $\mathcal{B}$. Assuming that this triad is positively oriented, its volume dV_0 in the reference configuration

$$\mathrm{dV}_0 = (\mathrm{d}\mathbf{X}^{(1)} \times \mathrm{d}\mathbf{X}^{(2)}) \cdot \mathrm{d}\mathbf{X}^{(3)} = \det(\mathrm{d}\mathbf{X}^{(1)}, \mathrm{d}\mathbf{X}^{(2)}, \mathrm{d}\mathbf{X}^{(3)}) \tag{2.7}$$

is given by the triple scalar product. In accordance with eq. (2.4) we can write

$$\mathrm{d}\mathbf{x}^{(i)} = \mathbf{F}\mathrm{d}\mathbf{X}^{(i)} \tag{2.8}$$

and can thus derive the volume

$$\mathrm{dV} = (\mathrm{d}\mathbf{x}^{(1)} \times \mathrm{d}\mathbf{x}^{(2)}) \cdot \mathrm{d}\mathbf{x}^{(3)} = \det(\mathbf{F}\mathrm{d}\mathbf{X}^{(1)}, \mathbf{F}\mathrm{d}\mathbf{X}^{(2)}, \mathbf{F}\mathrm{d}\mathbf{X}^{(3)}) \tag{2.9}$$

of the deformed element. Using the relationship

$$\det(\mathbf{AB}) = \det(\mathbf{A})\det(\mathbf{B}) \tag{2.10}$$

we can finally derive

$$\mathrm{dV} = \det(\mathbf{F})\det(\mathrm{d}\mathbf{X}^{(1)}, \mathrm{d}\mathbf{X}^{(2)}, \mathrm{d}\mathbf{X}^{(3)}) = \det(\mathbf{F})\mathrm{dV}_0 \equiv J\mathrm{dV}_0. \tag{2.11}$$

The determinant J of the deformation tensor (i.e. the Jacobian of the transformation φ) is the local ratio of current volume to reference volume of a material volume element. By definition (impenetrability of matter and non-singularity of $\mathbf{F}$),

$$J \equiv \det\mathbf{F} > 0. \tag{2.12}$$

It is also important to point out, especially in the context of soft tissue modeling, that for incompressible materials $J = 1$.
We now seek to describe the deformation of the infinitesimal surface element dA_0 with the normal $\mathbf{N}$ from its reference configuration $\mathrm{d}\mathbf{A}_0 = \mathrm{dA}_0\mathbf{N}$ to its current configuration $\mathrm{d}\mathbf{A} = \mathrm{dA}\mathbf{n}$. We start by expressing the volume of a deformed parallelepiped eq. (2.9)

$$\mathrm{dV} = \mathrm{d}\mathbf{A} \cdot \mathrm{d}\mathbf{x} = J\mathrm{dV}_0 = J\mathrm{d}\mathbf{A}_0 \cdot \mathrm{d}\mathbf{X} \tag{2.13}$$

through its base area. Noting that

$$d\mathbf{A} \cdot d\mathbf{x} = d\mathbf{A} \cdot \mathbf{F} d\mathbf{X} = \mathbf{F}^T d\mathbf{A} \cdot d\mathbf{X}, \tag{2.14}$$

we can derive

$$\underbrace{(\mathbf{F}^T d\mathbf{A} - J d\mathbf{A}_0)}_{0} \cdot d\mathbf{X} = 0 \tag{2.15}$$

As this must hold for arbitrary line elements $d\mathbf{X}$, we can describe the deformation of the arbitrary surface element $d\mathbf{A}_0$ through the deformation gradient tensor $\mathbf{F}$:

$$d\mathbf{A} = J\mathbf{F}^{-T} d\mathbf{A}_0 \tag{2.16}$$

This relationship is known as Nanson's formula.

2.2.3. Strain measures

The deformation gradient tensor characterizes the deformation of infinitesimal line, area and volume elements during the body motion. In order to construct meaningful material laws it is necessary to determine the strain (i.e. the 3D equivalent of stretch) inside a body. Strain can be described as a measure for the change in length of infinitesimal line elements. It should be pointed out that strain is not a necessarily a physically measurable quantity, but rather a theoretical concept. Consequently, many different strain measures exist. In the following, the most important strain tensors for soft tissue modeling are presented.
First, it is important to understand that the deformation gradient tensor is not a suitable strain tensor; this can be seen from the polar decomposition theorem.

Theorem 1 *The polar decomposition theorem: For any non-singular second order tensor* $\mathbf{A}$ *there exist a unique symmetric, positive definite second order tensor* $\mathbf{U}$ *and an orthogonal second-order tensor* $\mathbf{R}$ *such that*

$$\mathbf{A} = \mathbf{R}\mathbf{U} \tag{2.17}$$

For a proof we refer to Ogden [Ogd97]. With respect to the deformation gradient tensor (keep in mind that it is non-singular) this means that it can be decomposed into a pure rotation matrix $\mathbf{R}$ and a pure stretch matrix $\mathbf{U}$. Thus, $\mathbf{F}$ changes under pure rigid body motions. For apparent reasons, the invariance under rigid body motions is an important property of a useful strain measure. Thus $\mathbf{F}$ cannot be directly used as a strain tensor. One possibility to recover rotational invariance is to perform a polar decomposition for each computation step and to use the remaining stretch matrix as a strain measure. Alternatively, a quadratic strain measure can be used. The first approach is often used in real-time simulations (see chapter **??**), while the second possibility is used in classical solid mechanics as it allows for a better analytic analysis of the ensuing equations.
Upon inserting the deformation gradient tensor into the Cauchy-Green strain tensor

$$\mathbf{C} = \mathbf{F}^T\mathbf{F} \tag{2.18}$$

it is easy to see that $\mathbf{C}$ is rotation invariant (keep in mind that $\mathbf{R}^T\mathbf{R} = \mathbf{I}$ due to the orthogonality of $\mathbf{R}$):

$$\mathbf{C} = \mathbf{F}^T\mathbf{F} = (\mathbf{R}\mathbf{U})^T\mathbf{R}\mathbf{U} = \mathbf{U}^T\mathbf{R}^T\mathbf{R}\mathbf{U} = \mathbf{U}^T\mathbf{U}. \tag{2.19}$$

Another important strain measure is the related Green-Lagrange strain tensor

$$\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I}) = \frac{1}{2}(\mathbf{F}^T\mathbf{F} - \mathbf{I}) = \frac{1}{2}((\nabla\mathbf{u} + \mathbf{I})^T(\nabla\mathbf{u} + \mathbf{I}) - \mathbf{I}), \tag{2.20}$$

which is also symmetric and rotation invariant. Please note that both tensors are non-linear and must be in order to be rotation invariant. In order to facilitate the development of linear elasticity theory later on, we list the infinitesimal strain tensor

$$\boldsymbol{\epsilon} = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^T), \tag{2.21}$$

which is the linearization of the Green-Lagrange strain tensor.

2.2.4. Balance principles

In classical continuum mechanics, the behavior of objects is governed by four conservation laws: The conservation of mass, the balance of linear and angular momentum as well as the conservation of energy.
It is intuitively clear, that the body $\mathcal{B}$ is a closed system and its mass m does not change, even if $\mathcal{B}$ does occupy different geometrical regions $\Omega_0,...,\Omega$ over time. If we denote the density of $\mathcal{B}$ with $\rho_0(\mathbf{X})$ in the reference configuration and the density in the current configuration with $\rho(\mathbf{x})$ we thus require the mass to be the same in the current and in the reference configuration:

$$m = \int_{\Omega_0} \rho_0(\mathbf{X}) \mathrm{dV}_0 = \int_{\Omega} \rho(\mathbf{x}) \mathrm{dV} = \text{const.} > 0 \tag{2.22}$$

This is the conservation of mass in integral form. The linear momentum in the current configuration

$$\int_{\Omega} \rho \mathbf{v} \mathrm{dV} = \int_{\Omega} \rho \dot{\mathbf{x}} \mathrm{dV} \tag{2.23}$$

is changed, when $\mathcal{B}$ is subjected to external forces. In this context, so called volumetric body forces (e.g. gravity, electromagnetic forces) are distinguished from contact forces that act on the surface of $\mathcal{B}$. The gravitational body force

$$\int_{\Omega} \rho \mathbf{g} \mathrm{dV} \tag{2.24}$$

can be written in its integral form using the gravitational constant $\mathbf{g}$. An analogous integral description for the contact force

$$\int_{\partial\Omega} \mathbf{t}(\mathbf{x}, \partial\Omega) \mathrm{d}\mathbf{A} \tag{2.25}$$

can be found by defining the contact force density (or surface traction vector) $\mathbf{t}(\mathbf{x}, \partial\Omega)$. The balance of linear momentum can be written as

$$\int_{\partial\Omega} \mathbf{t}(\mathbf{x}, \partial\Omega) \mathrm{d}\mathbf{A} + \int_{\Omega} \rho \mathbf{g} \mathrm{dV} \equiv \frac{\mathrm{D}}{\mathrm{Dt}} \int_{\Omega} \rho \mathbf{v} \mathrm{dV} = \int_{\Omega} \rho \dot{\mathbf{v}} \mathrm{dV}. \tag{2.26}$$

For a body to be in complete equilibrium it is not sufficient that all internal and external forces acting on the body are in balance. Even if all external forces cancel out (static equilibrium) the body can still be subjected to a rotational motion, if these forces act on different points of the body. The rotational equilibrium is ensured by the balance of rotational momentum. By using the position vector $\mathbf{r}(\mathbf{x}) = \mathbf{x} - \mathbf{x}_0$ relative to a fixed point $\mathbf{x}_0$ it can be formulated as

$$\int_{\partial\Omega} \mathbf{r} \times \mathbf{t}(\mathbf{x}, \partial\Omega) \mathrm{d}\mathbf{A} + \int_{\Omega} \mathbf{r} \times \rho \mathbf{g} \mathrm{dV} \equiv \frac{\mathrm{D}}{\mathrm{Dt}} \int_{\Omega} \mathbf{r} \times \rho \mathbf{v} \mathrm{dV} = \int_{\Omega} \mathbf{r} \times \rho \dot{\mathbf{v}} \mathrm{dV}. \tag{2.27}$$

In the context of thermodynamics the conservation of energy and the balance of momentum is supplemented by the conservation of energy. However, in the realm of soft tissue simulation the thermal effects of the body motion are extremely small and are usually neglected. Thus, the body motion can be fully described using the three balance principles described above.

2.2.5. The concept of stress

Cauchy postulated that the surface traction vector $\mathbf{t}$ has the same value for all boundaries with the same normal direction $\mathbf{n}$. In other words, $\mathbf{t}$ only depends on the surface normal and we can write:

$$\mathbf{t}(\mathbf{x}, \partial\Omega) = \mathbf{t}(\mathbf{x}, \mathbf{n}). \tag{2.28}$$

From this postulate, Cauchy's fundamental stress theorem can be established.

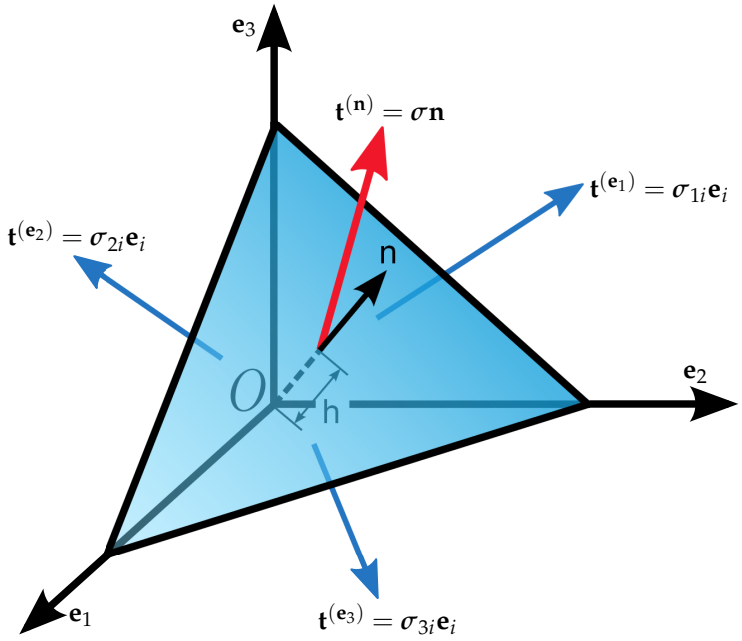


Figure 2.2.: Cauchy's tetrahedron: The traction $\mathbf{t}^{(\mathbf{n})}$ on the surface with normal $\mathbf{n}$ can be expressed as a linear combination of the tractions on the coordinate planes (based on [Wik13]).

Theorem 2 *Cauchy's stress theorem: Provided that it is continuous in* $\mathbf{x}$*, the stress vector* $\mathbf{t}(\mathbf{x},\mathbf{n})$ *depends linearly on* $\mathbf{n}$*, i.e. there exists a second order tensor field* σ *independent of* $\mathbf{n}$*, such that*

$$\mathbf{t}(\mathbf{x},\mathbf{n}) = \sigma(\mathbf{x})\mathbf{n} \tag{2.29}$$

for all $\mathbf{x}$ *in* $\mathcal{B}$*. The tensor* σ *is called the Cauchy stress (or true stress) tensor.*

The theorem is usually proven ([Ogd97], [Bon97]) by constructing a tetrahedron that lies in the Cartesian rectangular planes (see Fig. 2.2). After applying the balance of linear momentum eq. (2.26) and collapsing the height of the triangle ($h \mapsto 0$) the external body and inertia forces vanish. The application of the postulate then leads to Cauchy's stress tensor. Furthermore, it can be shown that the balance of rotational momentum implies the symmetry of the Cauchy stress tensor [HG01].
As a direct consequence we can express the force

$$\mathrm{d}\mathbf{f} = \mathbf{t}(\sigma,\mathbf{n})\mathrm{d}\mathbf{A} = \sigma(\mathbf{x})\mathbf{n}\mathrm{dA} \tag{2.30}$$

on an infinitesimal surface element (the so called surface tractions) in the current configuration using the Cauchy stress tensor and the surface normal.
Please note that the balance laws have so far been formulated in the current, deformed configuration. However, in a typical scenario the deformed geometry of $\mathcal{B}$ is actually the solution that we would like to solve for. It is thus impossible to integrate over the current configuration. This problem can be overcome by relating all forces to the reference configuration and solving the problem using the so called material description. In order to facilitate this formulation, we relate the surface force $\mathrm{d}\mathbf{f}$ to the undeformed surface element $\mathrm{d}\mathbf{A}_0$ through the use of Nanson's formula eq. (2.16):

$$\mathrm{d}\mathbf{f} = \sigma(\mathbf{x})\mathbf{n}\mathrm{dA} = \sigma J\mathbf{F}^{-T}\mathbf{N}\mathrm{dA}_0 = \mathbf{P}\mathbf{N}\mathrm{dA}_0 \tag{2.31}$$

Here, we introduced the first Piola-Kirchhoff stress tensor

$$\mathbf{P} = J\sigma\mathbf{F}^{-T} \tag{2.32}$$

which relates surface forces in the current configuration to surface elements in the reference configuration. The passage from σ to $\mathbf{P}$ is often referred to as the Piola transformation. Please note that, in contrast to the Cauchy stress tensor, the first Piola-Kirchhoff tensor is not symmetric (because $\mathbf{F}$ is generally not symmetric).
Many different stress measures have been proposed in the literature apart from the Cauchy and the first Piola-Kirchhoff stress tensor. At this point we will only mention the symmetric second Piola-Kirchoff stress tensor

$$\mathbf{S} = J\mathbf{F}^{-1}\sigma\mathbf{F}^{-T} = \mathbf{F}^{-1}\mathbf{P} = \mathbf{S}^T \tag{2.33}$$

which is very important in the context of soft tissue simulations for reasons that will be extensively discussed in chapter 2.3.

2.2.6. Boundary value problem of elasticity

The conservations laws can be combined into a single partial differential equation (PDE). Together with appropriate boundary conditions and a material law, this PDE forms a boundary value problem. We start by inserting the Cauchy stress tensor into the balance of linear momentum (eq. 2.26) to derive

$$\int_{\partial\Omega} \sigma(\mathbf{x})\mathbf{n}\mathrm{dA} + \int_{\Omega} \rho\mathbf{g}\mathrm{dV} = \int_{\Omega} \rho\dot{\mathbf{v}}\mathrm{dV}. \tag{2.34}$$

From this, Cauchy's first equation of motion

$$\int_{\Omega} \mathrm{div}\sigma(\mathbf{x})\mathrm{dV} + \int_{\Omega} \rho\mathbf{g}\mathrm{dV} = \int_{\Omega} \rho\dot{\mathbf{v}}\mathrm{dV} \tag{2.35}$$

is derived by applying the divergence theorem (A.1.1) to the surface term. As this relation has to hold for any volume dV in $\mathcal{B}$ the differential (local) form

$$\mathrm{div}\sigma(\mathbf{x}) + \rho\mathbf{g} = \rho\dot{\mathbf{v}} \tag{2.36}$$

immediately follows from the integral (global) form (2.35).
As stated above, this partial differential equation cannot be solved when formulated in terms of the current (unknown) configuration (spatial description). Therefore, we use the mass conservation and Nanson's formula to express eq. (2.34) with respect to the reference configuration:

$$\int_{\partial\Omega_0} \sigma J\mathbf{F}^{-T}\mathbf{N}\mathrm{dA}_0 + \int_{\Omega_0} \rho_0\mathbf{g}\mathrm{dV}_0 = \int_{\Omega_0} \rho_0\dot{\mathbf{v}}\mathrm{dV}_0. \tag{2.37}$$

Inserting the definition of the first Piola-Kirchhoff stress tensor eq. (2.32) and application of the divergence theorem yields the material description of Cauchy's first equation of motion

$$\int_{\Omega_0} \mathrm{Div}\mathbf{P}\mathrm{dV}_0 + \int_{\Omega_0} \rho_0\mathbf{g}\mathrm{dV}_0 = \int_{\Omega_0} \rho_0\dot{\mathbf{v}}\mathrm{dV}_0. \tag{2.38}$$

The boundary value problem is typically stated using the local formulation

$$\mathrm{Div}\mathbf{P} + \rho_0\mathbf{g} = \rho_0\dot{\mathbf{v}} \qquad \forall\mathbf{x} \in \Omega_0 \tag{2.39}$$

of Cauchy's equation of motion in material description. In addition to the equilibrium equation, boundary conditions have to be specified on the elastic body $\mathcal{B}$ in order to pose a physically sensible problem (please see Fig. 2.3 for an example). The parts of the surface $\Gamma_D \subseteq \partial\Omega$ where the position (or the displacement) of the body are known are called Dirichlet boundary conditions. In contrast, surface tractions are imposed on the Neumann boundary Γ_N. In order for the problem to be well posed, either Dirichlet or Neumann boundary conditions have to be prescribed on the whole boundary ($\partial\Omega = \Gamma_D \cup \Gamma_N$). Furthermore, Γ_D and Γ_N are not allowed to overlap, i.e. $\Gamma_D \cap \Gamma_N = \emptyset$. By specifying the spaces of functions that satisfy these boundary conditions

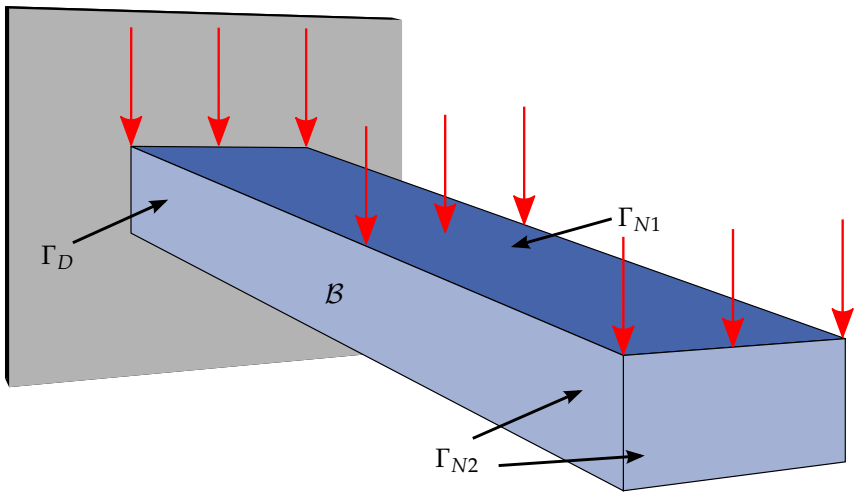


Figure 2.3.: A cantilever beam is fixed at the left end (zero displacement at Dirichlet boundary Γ_D) and a uniform surface pressure is applied at the top (on Neumann boundary Γ_{N1}). Zero force boundary conditions are prescribed on all other surfaces (Neumann boundary Γ_{N2}).

$$V_D(\Gamma_D, \bar{\mathbf{x}}) = \{\mathbf{x} | \mathbf{x} = \bar{\mathbf{x}} \qquad \forall \mathbf{x} \in \Gamma_D\} \tag{2.40}$$
$$V_N(\Gamma_N, \bar{\mathbf{t}}) = \{\mathbf{x} | (\sigma \mathbf{n}) = \bar{\mathbf{t}} \qquad \forall \mathbf{x} \in \Gamma_N\} \tag{2.41}$$

we can finally state the boundary value problem of non-linear elasticity:
Find $\mathbf{x} \in C^2(\Omega) \cap C^1(\overline{\Omega}) \cap V_D(\Gamma_D, \bar{\mathbf{x}}) \cap V_N(\Gamma_N, \bar{\mathbf{t}})$ s.t. eq. 2.39 holds.

2.3. Material laws for biological soft tissue

The boundary value problem introduced in the previous section cannot be solved without a constitutive equation that relates the position $\mathbf{x}$ (or the displacement $\mathbf{u}$) to the stress tensor values. In the context of elastic bodies this relationship is called the response function $\mathcal{G}(\mathbf{F}(\mathbf{X}, t), \mathbf{X})$. In the following section we will see that some of the already encountered strain and stress measures form special (so called work conjugate) pairs. The work conjugancy relationship arises from the definition of the internal elastic energy that is stored in $\mathcal{B}$ during the deformation. Furthermore we will see that the internal elastic energy is an important concept in the context of hyperelastic material models, which are the most important class of non-linear models for soft tissue mechanics. We will also introduce basic techniques for modeling viscoelastic behavior. Finally, it is shown under which assumptions the nonlinear elasticity problem reduces to a linear problem.

2.3.1. Mechanical energy in elastic bodies

If the material response is purely elastic and no energy is dissipated as heat, the balance of mechanical energy can be directly derived from the equation of motion. Multiplying Cauchy's equation of motion (2.36) with the velocity $\mathbf{v}$ yields

$$\mathrm{div}\sigma \cdot \mathbf{v} + \rho \mathbf{g} \cdot \mathbf{v} = \rho \dot{\mathbf{v}} \cdot \mathbf{v}. \tag{2.42}$$

By using the product rule (see A.1.2) and introducing the spatial velocity gradient $\mathbf{l} = \mathrm{grad}\mathbf{v}$ we derive

$$\mathrm{div}(\sigma \mathbf{v}) - \sigma : \mathbf{l} + \rho \mathbf{g} \cdot \mathbf{v} = \rho \dot{\mathbf{v}} \cdot \mathbf{v}. \tag{2.43}$$

We note that in light of the product differentiation rule, the mass term can be rewritten to

$$\rho\dot{\mathbf{v}}\cdot\mathbf{v}=\rho\frac{1}{2}\overline{\mathbf{v}\cdot\mathbf{v}}^{\cdot}=\rho\frac{\mathrm{D}}{\mathrm{D}t}\frac{1}{2}\mathbf{v}^2. \tag{2.44}$$

The spatial velocity gradient $\mathbf{l}$ is usually additively decomposed

$$\mathbf{l}=\mathbf{d}+\mathbf{w} \tag{2.45}$$

into the symmetric rate of deformation tensor

$$\mathbf{d}=\frac{1}{2}(\mathbf{l}+\mathbf{l}^T)=\mathbf{d}^T \tag{2.46}$$

and the antisymmetric rate of rotation sensor

$$\mathbf{w}=\frac{1}{2}(\mathbf{l}-\mathbf{l}^T)=-\mathbf{w}^T. \tag{2.47}$$

It is quickly shown that the material velocity gradient

$$\mathrm{Grad}\mathbf{v}=\frac{\partial\mathbf{v}(\mathbf{X},t)}{\partial\mathbf{X}}=\frac{\mathrm{D}}{\mathrm{D}t}\frac{\partial\varphi(\mathbf{X},t)}{\partial\mathbf{X}}=\dot{\mathbf{F}} \tag{2.48}$$

is identical to the time rate change $\dot{\mathbf{F}}$ of the deformation gradient. The relationship between the spatial velocity gradient $\mathbf{l}$ and $\dot{\mathbf{F}}$ is given by

$$\mathbf{l}=\frac{\partial\mathbf{v}}{\partial\mathbf{x}}=\frac{\mathrm{D}}{\mathrm{D}t}\frac{\partial\varphi\ (\mathbf{X},t)}{\partial\mathbf{x}}=\frac{\mathrm{D}}{\mathrm{D}t}\frac{\partial\varphi\ (\mathbf{X},t)}{\partial\mathbf{X}}\frac{\partial\mathbf{X}}{\partial\mathbf{x}}=\dot{\mathbf{F}}\mathbf{F}^{-1}. \tag{2.49}$$

We note that due to the symmetry of $\boldsymbol{\sigma}$, we have

$$\boldsymbol{\sigma}:\mathbf{l}=\boldsymbol{\sigma}:\mathbf{d}+\boldsymbol{\sigma}:\mathbf{w}=\boldsymbol{\sigma}:\mathbf{d}. \tag{2.50}$$

Inserting this result into eq. (2.43) as well as using the relationship (2.45) and subsequently integrating over the volume of the current configuration yields

$$\frac{\mathrm{D}}{\mathrm{D}t}\int_\Omega\frac{1}{2}\rho\mathbf{v}^2\mathrm{d}V+\int_\Omega\boldsymbol{\sigma}:\mathbf{d}\mathrm{d}V=\int_\Omega\mathrm{div}(\boldsymbol{\sigma}\mathbf{v})\mathrm{d}V+\int_\Omega\rho\mathbf{g}\cdot\mathbf{v}\mathrm{d}V. \tag{2.51}$$

The balance equation for mechanical energy in spatial description

$$\frac{\mathrm{D}}{\mathrm{D}t}\int_\Omega\frac{1}{2}\rho\mathbf{v}^2\mathrm{d}V+\int_\Omega\boldsymbol{\sigma}:\mathbf{d}\mathrm{d}V=\int_{\partial\Omega}\mathbf{t}\cdot\mathbf{v}\mathrm{d}A+\int_\Omega\rho\mathbf{g}\cdot\mathbf{v}\mathrm{d}V \tag{2.52}$$

immediately follows from the application of the divergence theorem and the definition of the surface traction (2.30). The right hand side of the equilibrium equation is the external mechanical power or rate of external mechanical work

$$\mathcal{P}_{ext}(t)=\int_{\partial\Omega}\mathbf{t}\cdot\mathbf{v}\mathrm{d}\mathbf{A}+\int_\Omega\rho\mathbf{g}\cdot\mathbf{v}\mathrm{d}V \tag{2.53}$$

is the power input on the region Ω. The kinetic energy

$$\mathcal{K}(t)=\int_\Omega\frac{1}{2}\rho\mathbf{v}^2\mathrm{d}V \tag{2.54}$$

can be regarded as an generalization of Newtonian mechanics to continuum mechanics. If $\mathcal{K}$ is zero (i.e. no inertia forces), then the dynamic BVP reduces to a non-linear static problem. The stress power or rate of internal mechanical work is given by

$$\mathcal{P}_{int}(t)=\int_\Omega\boldsymbol{\sigma}:\mathbf{d}\mathrm{d}V. \tag{2.55}$$

In order to derive the balance of mechanical energy in material (Lagrangian) form, we formulate the internal mechanical work in terms of the material description:

$$\mathcal{P}_{int}(t) = \int_{\Omega_0} \boldsymbol{\sigma} : \mathbf{1} J \mathrm{dV}_0 = \int_{\Omega_0} J\boldsymbol{\sigma} : \dot{\mathbf{F}}\mathbf{F}^{-1} \mathrm{dV}_0 = \int_{\Omega_0} J\mathrm{tr}(\boldsymbol{\sigma}^T \dot{\mathbf{F}}\mathbf{F}^{-1}) \mathrm{dV}_0 \tag{2.56}$$

$$= \int_{\Omega_0} J\mathrm{tr}(\dot{\mathbf{F}}\mathbf{F}^{-1}\boldsymbol{\sigma}) \mathrm{dV}_0 = \int_{\Omega_0} J\mathrm{tr}((\dot{\mathbf{F}}\mathbf{F}^{-1}\boldsymbol{\sigma})^T) \mathrm{dV}_0 \tag{2.57}$$

$$= \int_{\Omega_0} J\mathrm{tr}(\boldsymbol{\sigma}^T \mathbf{F}^{-T} \dot{\mathbf{F}}^T) \mathrm{dV}_0 = \int_{\Omega_0} J\boldsymbol{\sigma}\mathbf{F}^{-T} : \dot{\mathbf{F}} \mathrm{dV}_0 = \int_{\Omega_0} \mathbf{P} : \dot{\mathbf{F}} \mathrm{dV}_0 \tag{2.58}$$

We furthermore define the first Piola-Kirchhoff traction vector

$$\mathbf{T} \mathrm{d}\mathbf{A}_0 = \mathbf{t} \mathrm{d}\mathbf{A} \tag{2.59}$$

in order establish the balance of mechanical energy in material description:

$$\frac{\mathrm{D}}{\mathrm{Dt}} \int_{\Omega_0} \frac{1}{2} \rho_0 \mathbf{v}^2 \mathrm{dV}_0 + \int_{\Omega_0} \mathbf{P} : \dot{\mathbf{F}} \mathrm{dV}_0 = \int_{\partial\Omega_0} \mathbf{T} \cdot \mathbf{v} \mathrm{d}\mathbf{A}_0 + \int_{\Omega_0} \rho_0 \mathbf{g} \cdot \mathbf{v} \mathrm{dV}_0 \tag{2.60}$$

The stress power per unit reference volume

$$w_{int} = J\boldsymbol{\sigma} : \mathbf{d} = \mathbf{P} : \dot{\mathbf{F}} = \mathbf{S} : \dot{\mathbf{E}} \tag{2.61}$$

of a material is thus given by the double contraction of a stress tensor and an associated strain rate tensor. The equation above lists the most important ones of these couples known as work conjugate pairs (please refer to the appendix A.2 on how to derive the work conjugacy of the second Piola-Kirchhoff stress tensor $\mathbf{S}$ and the material time derivative of the Green-Lagrange strain tensor $\dot{\mathbf{E}}$).

2.3.2. Hyperelastic materials

Biological soft tissue is usually modeled using a phenomenological approach. Based on *in vitro* or *in vivo* measurements, mathematical models are fitted to experimental data that describe the stress-strain relationship. In this section, the important hyperelastic approach to soft tissue modeling is discussed. Although only isotropic hyperelastic materials are considered at this point, the approach can be extended to include anisotropic material behavior [HG01]. The extension of the model to viscoelastic material response is detailed in the subsequent section.

Materials are called Cauchy-elastic if the stress field in the deformed configuration only depends on the state of deformation and not on the deformation history. That means we can define a so called response function $\mathcal{G}$ that relates the deformation gradient tensor $\mathbf{F}$ to the Cauchy stress field $\boldsymbol{\sigma}$:

$$\boldsymbol{\sigma}(\mathbf{x}, t) = \mathcal{G}(\mathbf{F}(\mathbf{X}, t), \mathbf{X}) \tag{2.62}$$

By inserting this definition into the already familiar Piola transformation

$$\mathbf{P} = J\boldsymbol{\sigma}\mathbf{F}^{-T} = J\mathcal{G}(\mathbf{F})\mathbf{F}^{-T} = \mathcal{H}(\mathbf{F}) \tag{2.63}$$

we can define the response function $\mathcal{H}$ that relates the deformation to the first Piola-Kirchhoff stress tensor.

The problem of describing a suitable response function for biological soft tissue is usually tackled by describing the internal energy of a material. For this purpose, we postulate the existence of a so called elastic potential or strain energy function Ψ that is defined per unit reference volume. Materials for which Ψ exists and only depends on the deformation ($\Psi = \Psi(\mathbf{F})$) and not on the deformation history are called (pure) hyperelastic materials.

We now show how elastic response functions for work conjugate stress-strain tensors pairs can be derived from the elastic potential. The time derivative of the internal energy

$$\dot{\Psi} = w_{int} = \mathbf{P} : \dot{\mathbf{F}} \tag{2.64}$$

is the internal work which can be expressed through the work conjugate pair. On the other hand, we can apply the chain rule to derive

$$\dot{\Psi} = \frac{\partial \Psi}{\partial \mathbf{F}} : \dot{\mathbf{F}}. \tag{2.65}$$

By subtracting the above equations we obtain

$$\left(\frac{\partial \Psi}{\partial \mathbf{F}} - \mathbf{P} \right) : \dot{\mathbf{F}} = 0 \tag{2.66}$$

which has to hold for arbitrary $\mathbf{F}$ and $\dot{\mathbf{F}}$ and thus we can conclude:

$$\mathbf{P} = \frac{\partial \Psi}{\partial \mathbf{F}} = \mathcal{H}(\mathbf{F}) \tag{2.67}$$

The same technique can be used to express the second Piola-Kirchhoff stress tensor

$$\mathbf{S} = \frac{\partial \Psi}{\partial \mathbf{E}} = \mathcal{H}(\mathbf{E}) \tag{2.68}$$

in terms of the Green-Lagrange strain tensor. This relationship allows to compute the response function once the relationship between the elastic potential and the deformation is known. Please note that while hyperelastic materials (also called Green-elastic) are evidently always Cauchy-elastic, the converse is not necessarily true. Although the stress field for Cauchy-elastic materials is independent of the deformation path, in contrast to hyperleastic materials the work done (i.e. the internal energy) by the stress field can depend on the deformation history.
Naturally, material models should be constructed in a way that the boundary value problem has an (ideally unique) solution that corresponds to the physical observations. The mathematical treatment of the uniqueness and existence of non-linear elasticity problems is still an area of active research and revolves around the concept of the polyconvexity of strain-energy functions (see e.g. [HN03] [H$^+$94]).
On the physical level, there is one important necessary condition for the strain energy function: It should be invariant under superimposed rigid-body motions, i.e.

$$\Psi(\mathbf{F}) = \Psi(\mathbf{QF}) \tag{2.69}$$

for all orthogonal tensors $\mathbf{Q}$. If we choose $\mathbf{Q}$ to be the transpose of the orthogonal rotation tensor $\mathbf{R}$ that arises from the polar decomposition (see theorem 1) of $\mathbf{F}$,

$$\Psi(\mathbf{F}) = \Psi(\mathbf{R}^T\mathbf{RU}) = \Psi(\mathbf{U}) \tag{2.70}$$

we learn that Ψ has to be independent from the rotational component $\mathbf{R}$ in order to be invariant under superimposed rigid-body motions. In classical continuum mechanics, the strain energy is usually expressed in terms of the quadratic, rotation invariant Cauchy-Green deformation tensor $\mathbf{C}$ (or the Green-Lagrange tensor $\mathbf{E}$) instead of the pure stretch tensor $\mathbf{U}$. Thus it is not necessary to perform a polar decomposition during analysis.
A material is called isotropic if its properties (e.g. the stress response) are identical in all directions. This means, that the property is not affected if the reference configuration is translated or rotated. If the strain energy function is formulated in terms of the Cauchy-Green deformation tensor $\mathbf{C}$, this requirement can be expressed mathematically as

$$\Psi(\mathbf{C}) = \Psi(\mathbf{QCQ}^T) \tag{2.71}$$

where $\mathbf{Q}^T$ again denotes an arbitrary orthogonal tensor (rotation matrix). The representation theorem of invariants shows how to construct the strain energy function for isotropic materials [Hol00]:

Theorem 3 *The representation theorem for invariants: If a scalar-valued tensor function with the argument* $\mathbf{C}$ *is an invariant under a rotation according to (2.71), it may be expressed in terms of the principal invariants of* $\mathbf{C}$*:*

$$I_1(\mathbf{C}) = \mathrm{tr}\mathbf{C} \tag{2.72}$$

$$I_2(\mathbf{C}) = \frac{1}{2}\left[(\mathrm{tr}\mathbf{C})^2 - \mathrm{tr}(\mathbf{C})^2\right] \tag{2.73}$$

$$I_3(\mathbf{C}) = \det\mathbf{C} \tag{2.74}$$

The questions remains how the strain energy function should be constructed. If Ψ is continuously differentiable with respect to the invariants, we can expand Ψ into the infinite power series

$$\Psi(I_1, I_2, I_3) = \sum_{p,g,r=0}^{\infty} c_{pgr}(I_1 - 3)^p (I_2 - 3)^q (I_3 - 1)^r. \tag{2.75}$$

Here, the coefficients c_{pqr} are the material parameters that have to be experimentally determined. Please note that this expansion has been chosen such that the material is energy-free in the reference configuration (i.e. $\mathbf{C} = \mathbf{I}$). It is a common approach to separate the strain energy functional

$$\Psi = \Psi_{iso} + \Psi_{vol} \tag{2.76}$$

into a part Ψ_{vol} that only depends on the volume change and a so called isochoric part Ψ_{iso} that is independent of the volumetric changes [Hol00].
We have already discovered in section 2.2.2 that $I_3 = \det\mathbf{C} = (\det\mathbf{F})^2$ is a measure of the volumetric change during the deformation. Therefore, it is readily seen that the general expression

$$\Psi_{vol} = \sum_{r=0}^{\infty} c_r (I_3 - 1)^r \tag{2.77}$$

describes an internal energy that is induced by volumetric changes. A simple and yet widely used formulation is

$$\Psi_{vol} = p(I_3 - 1)^2. \tag{2.78}$$

In the fully incompressible case, p serves a Lagrange multiplier during the computation of the solution and can be associated with the hydrostatic pressure. In this case, p is not a material parameter, but can be determined through the incompressibility constraint. If the material is modeled as nearly incompressible (which is often the case for biological soft tissue), p can be regarded as a penalty factor for the volumetric change. In this case, it is often replaced through its inverse $D_1 = 1/p$.
If the material is incompressible, there are no volume changes and the isochoric part of the strain energy

$$\Psi_{iso}(\mathbf{C}) = \Psi_{iso}(I_1, I_2) \tag{2.79}$$

depends on the invariants I_1, I_2. However, these invariants vary during volumetric changes. In the context of compressible materials, the modified deformation tensor $\overline{\mathbf{F}} = J^{-1/3}\mathbf{F}$ and the associated modified right Cauchy-Green tensor $\overline{\mathbf{C}} = \overline{\mathbf{F}}^T\overline{\mathbf{F}}$ are used as deformation measures in the strain energy function. It is evident, that $\det\overline{\mathbf{F}} = \det\overline{\mathbf{C}} = 1$ and thus the strain energy

$$\Psi_{iso}(\overline{\mathbf{C}}) = \Psi_{iso}(\overline{I}_1, \overline{I}_2) \tag{2.80}$$

based on the invariants $\overline{I}_1, \overline{I}_2$ of $\overline{\mathbf{C}}$ is not influenced by volumetric changes.
In the following section, common material models are presented for the isochoric strain energy. Although we will formulate them for the incompressible case in the form of the potential $\Psi_{iso}(\mathbf{C})$, it is important to note that they generalize to the compressible domain by simply using the modified formulation $\Psi_{iso}(\overline{\mathbf{C}})$ based on the modified deformation measures introduced above [Hol00].
The Mooney-Rivlin material model

$$\Psi_{iso}(\mathbf{C}) = c_1(I_1 - 3) + c_2(I_2 - 3) \tag{2.81}$$

has originally been developed for isotropic rubber-like materials and is often used for soft tissue modeling. One of the simplest hyperelastic models is the neo-Hookean model

$$\Psi_{iso}(\mathbf{C}) = c_1(I_1 - 3). \tag{2.82}$$

The material parameter c_1 can be associated with the shear modulus μ by the formula $\mu = 2c_1$. The neo-Hookean model can be considered a special case of the reduced polynomial model

$$\Psi_{iso}(\mathbf{C}) = \sum_{i=1}^{N} c_i (I_1 - 3)^i \tag{2.83}$$

with $N = 1$ [RES10].
The simplest model for a compressible hyperelastic material is the Saint Venant-Kirchhoff model. For simplicity reasons, its strain energy function

$$\Psi(\mathbf{E}) = \frac{\lambda}{2}(\mathrm{tr}\mathbf{E})^2 + \mu \mathrm{tr}(\mathbf{E}^2) \tag{2.84}$$

is usually formulated in terms of the Green-Lagrange strain tensor. The parameter λ is called Lame's first parameter, while μ denotes the shear modulus (or Lame's second parameter). The Saint-Venant Kirchhoff model is often used for real-time applications in computer graphics. It is quickly shown (see appendix A.3 for details) that this model results in the linear relationship

$$\mathbf{S} = \frac{\partial \Psi}{\partial \mathbf{E}} = \lambda(\mathrm{tr}\mathbf{E})\mathbf{I} + 2\mu\mathbf{E} \tag{2.85}$$

between $\mathbf{S}$ and $\mathbf{E}$. Although the model can be very well suited for many large displacement problems, its formulation has several disadvantages. It is not based on a decomposition of the strain energy function into an isochoric and volumetric part (the third invariant J is not even explicitly used). It is also not monotonic in compression and can thus break down for large compressive strains. Consequently it does not satisfy the polyconvexity condition.

2.3.3. Visco-elasticity

The stress response of a biological soft tissue does not only depend on the instantaneous strain, but also on the deformation history. This can be observed during a simple indentation experiment. The material response during the loading phase is different from the unloading (*recovery*) phase. In particular, the stress will only gradually decrease over time after the indenter has been completely removed. This relaxation process cannot be captured by purely hyperelastic models. A general approach to model this viscoelastic behavior is to express the time-dependent strain-energy function

$$\hat{\Psi} = \int_0^T G(t-s)\frac{\partial \Psi}{\partial s} ds \tag{2.86}$$

in terms of a convolution integral between the stress power and a relaxation function $G(t, \mathbf{C})$ [TCC+08]. In order to facilitate an efficient computation, it is often assumed that the relaxation function does not depend on the current strain. This approach to separate the purely hyperelastic material response from the viscoelastic behavior is called Quasi-Linear-Viscoelasticity (QLV) [Fun93]. A useful representation of the relaxation function $G(t)$ is given by the Prony series

$$G(t) = G_{\mathrm{inf}} + \sum_{i=1}^{N} G_i e^{-t/\tau_i} \tag{2.87}$$

where G_{inf} is the long term modulus once the material is totally relaxed and τ_i are relaxation times. This model has a physical interpretation in form of a special spring-dashpot network that is called the generalized Maxwell model [Hol00]: A spring with the stiffness G_{inf} is arranged in parallel to N Maxwell elements. Each Maxwell element in turn consists of a series of one spring and one dashpot. In practice it is often much more difficult to determine G_{inf} than the instantaneous (purely elastic) modulus G_0. By noting that

$$G(t=0) = G_0 = G_{\mathrm{inf}} + \sum_{i=1}^{N} G_i, \tag{2.88}$$

the relaxation function can be described in the equivalent form

$$G(t) = G_0 - \sum_{i=1}^{N} G_i(1 - e^{-t/\tau_i}). \tag{2.89}$$

If the relaxation function is normalized with G_0 we obtain

$$g(t) = 1 - \sum_{i=1}^{N} g_i (1 - e^{-t/\tau_i}) \tag{2.90}$$

and can subsequently express the time dependent hyperelastic material coefficients (see e.g. eq. 2.83) by the equation

$$\hat{c}_{ij}(t) = c_{ij} g(t). \tag{2.91}$$

The QLV model is considerably more computational intensive than a pure hyperelastic model. A more efficient, but less accurate approach is to use a phenomenological viscosity formulation based on the computational model. If a linear elastic model is discretized in space using the finite element method (we will discuss this procedure in detail in the next chapter) the result is the system of ordinary differential equations (ODEs)

$$\mathbf{M}\ddot{\mathcal{U}} + \mathbf{K}\mathcal{U} = \mathbf{f}^{ext}. \tag{2.92}$$

Here, $\mathcal{U}$ is a vector of nodal displacements, $\mathbf{M}$ denotes the mass matrix, $\mathbf{K}$ is the stiffness matrix and $\mathbf{f}^{ext}$ encapsulates the external forces. The idea of the widely used Rayleigh damping is to add an artificial damping term to the above equations

$$\mathbf{M}\ddot{\mathcal{U}} + \mathbf{D}\dot{\mathcal{U}} + \mathbf{K}\mathcal{U} = \mathbf{f}^{ext}. \tag{2.93}$$

In this formulation the damping matrix $\mathbf{D}$ is constructed by the linear combination

$$\mathbf{D} = \alpha\mathbf{M} + \beta\mathbf{K} \tag{2.94}$$

with the scalar coefficients α and β that control the viscoelastic behavior.
The generalization of the Rayleigh damping scheme to hyperelastic materials is straightforward. In this case, the discrete form eq. (2.92) is not a linear system of ordinary differential equations, but a non-linear one. Thus, the time-discretization of the ODEs yields a non-linear system of equations. This is typically solved by iteratively solving linear systems (Newton-Raphson algorithm). During this procedure, Rayleigh damping can be employed within each linearization step.

2.3.4. Linear elasticity

The non-linear system solve using Newton-Raphson iterations is not only computationally expensive, but can lead to instabilities especially for dynamic problems. If the deformation of the body is small (i.e. $\|\mathrm{Grad}\mathbf{u}\| << \mathbf{I}$) and the stress-strain relationship is linear, it is not necessary to solve the non-linear problem. Instead, a computationally efficient linear elasticity model can be used. It is important to point out that this model is often used for metals. However, soft tissue deformations are usually large and the small strain assumption is thus not justified. Also, as discussed above, the material response of soft tissue is highly non-linear. It can thus be expected (and will indeed be shown in study presented in the next section) that the linear model introduces a significant error in soft tissue simulation. However, we will see in the following chapter that linear elasticity serves as an important building block for real-time capable algorithms. Therefore, the model will be presented in this section. For a more mathematical rigorous derivation of the linear model from the presented non-linear, hyperelastic model we refer to the textbook by Ogden [Ogd97].
In the context of linear elasticity it is often more convenient to use the displacement $\mathbf{u}$ (see eq. 2.3) as the primary variable. Recalling the relationship between $\mathbf{F}$ and $\mathbf{u}$ eq. (2.5) we can derive

$$\mathrm{Grad}\mathbf{u} = \frac{\partial u_i}{\partial X_j} = \frac{\partial u_i}{\partial x_k}\frac{\partial x_k}{\partial X_j} = (\mathrm{grad}\mathbf{u})\mathbf{F} = \mathrm{grad}\mathbf{u}(\mathrm{Grad}\mathbf{u} + \mathbf{I}) \tag{2.95}$$

Thus it follows that under the small strain (linear) approximation $\mathrm{Grad}\mathbf{u}$ and $\mathrm{grad}\mathbf{u}$ can be used interchangeably. In other words, as the reference configuration and the deformed configuration are

nearly identical, i.e. the spatial and material derivative are the nearly the same. Consequently, the already defined infinitesimal strain tensor

$$\boldsymbol{\epsilon} = \frac{1}{2}(\mathrm{Grad}\mathbf{u} + (\mathrm{Grad}\mathbf{u})^T) = \frac{1}{2}(\mathrm{grad}\mathbf{u} + (\mathrm{grad}\mathbf{u})^T) \tag{2.96}$$

can be alternatively expressed through a spatial derivative. Under these approximations, the rate deformation tensor

$$\mathbf{d} = \frac{1}{2}(\mathrm{grad}\mathbf{v} + (\mathrm{grad}\mathbf{v})^T) \approx \dot{\boldsymbol{\epsilon}} \tag{2.97}$$

can be regarded as the time derivative of the infinitesimal strain tensor. This leads to the important finding that in the framework of linear elasticity, $\dot{\boldsymbol{\epsilon}}$ is work conjugated to the Cauchy stress tensor $\boldsymbol{\sigma}$ (see eq. 2.61). Under the small strain assumption the strain energy rate per unit reference volume can be approximated by the strain energy rate per unit deformed volume, i.e.

$$\dot{\Psi} = J\boldsymbol{\sigma} : \dot{\boldsymbol{\epsilon}} \approx \boldsymbol{\sigma} : \dot{\boldsymbol{\epsilon}}. \tag{2.98}$$

Through this result we can establish the material law using the hyperelastic strain energy based approach. As the purpose of the small strain approximation is to achieve a completely linear formulation, the Saint-Venant Kirchhoff model is the obvious choice. With the previous results eq. (2.85) we obtain

$$\boldsymbol{\sigma} = \frac{\partial \Psi}{\partial \boldsymbol{\epsilon}} = \lambda \mathrm{tr}\boldsymbol{\epsilon}\mathbf{I} + 2\mu\boldsymbol{\epsilon}. \tag{2.99}$$

In order to compactly state the linear elastic BVP we again define the function spaces that satisfy the boundary conditions

$$V_D(\Gamma_D, \overline{\mathbf{u}}) = \{\mathbf{u} | \mathbf{u} = \overline{\mathbf{u}} \qquad \forall \mathbf{u} \in \Gamma_D\} \tag{2.100}$$

$$V_N(\Gamma_N, \bar{\mathbf{t}}) = \{\mathbf{u} | (\boldsymbol{\sigma}\mathbf{n}) = \bar{\mathbf{t}} \qquad \forall \mathbf{u} \in \Gamma_N\} \tag{2.101}$$

on the Dirichlet boundary Γ_D and on the Neumann boundary Γ_N, respectively. By using the Cauchy stress in the spatial Cauchy equation of motion eq. (2.36), we can formulate the complete, displacement-based boundary value problem for linear elasticity: Find $\mathbf{u} \in C^2(\Omega) \cap C^1(\overline{\Omega}) \cap V_D(\Gamma_D, \overline{\mathbf{u}}) \cap V_N(\Gamma_N, \bar{\mathbf{t}})$ s.t.

$$\mathrm{Div}\boldsymbol{\sigma} + \rho_0\mathbf{g} = \rho_0\ddot{\mathbf{u}} \qquad \forall \mathbf{u} \in \Omega_0 \tag{2.102}$$

$$\boldsymbol{\epsilon} = \frac{1}{2}(\mathrm{Grad}\mathbf{u} + (\mathrm{Grad}\mathbf{u})^T). \tag{2.103}$$

It is very important to point out that in linear elasticity, the boundary value problem is defined in terms of the reference configuration. However, the Cauchy stress tensor is used instead of the first Piola-Kirchhoff tensor. The physical explanation for this approximation is that the reference configuration and the deformed configuration coincide and thus the Piola transform becomes unnecessary. More mathematically speaking, there exists one linear approximation of both the first Piola-Kirchhoff and the second Piola-Kirchhof stress tensor. This linear stress tensor can be identified as the Cauchy stress tensor (see Ogden for details [Ogd97]).
By noting that $\mathbf{v} = \dot{\mathbf{x}} = \dot{\mathbf{u}}$ we can also state the balance of mechanical energy for the linear elasticity formulation:

$$\frac{\mathrm{D}}{\mathrm{D}t}\int_{\Omega_0} \frac{1}{2}\dot{\mathbf{u}}^2 \mathrm{d}V + \int_{\Omega_0} \boldsymbol{\sigma} : \dot{\boldsymbol{\epsilon}} \mathrm{d}V = \int_{\partial\Omega_0} \mathbf{t} \cdot \dot{\mathbf{u}} \mathrm{d}\mathbf{A} + \int_{\Omega_0} \rho\mathbf{g} \cdot \dot{\mathbf{u}} \mathrm{d}V \tag{2.104}$$

A computer will do what you tell it to do, but that may be much different from what you had in mind.

— Joseph Weizenbaum

3.

Finite element based discretization

Unfortunately the boundary value problem (BVP) of elasticity can (with exception of a few simple cases) not be solved analytically. In this chapter, we outline a numerical approach for solving the linear BVP (2.103) based on the finite element method (FEM). It should be pointed out that the approach generalizes very well to the non-linear problem.

3.1. Getting Started

The core idea of the FE method is to determine the exact solution for the BVP not in a continuous (infinite-dimensional) space V, but rather in the discrete (finite-dimensional) sub-space $V_h \subset V$. In order to achieve that goal the partial differential equations of the BVP are transformed into a so called weak formulation. The weak formulation only requires the solution to hold only with respect to so called test functions. Although this concept seems odd at first (and much more complicated than the original partial differential equations), we will quickly see that it can be discretized very elegantly. We will also discover that the weak formulation is identical with the minimization of certain energy functionals (variational formulation). Especially in the static case this opens up a very vivid physical interpretation of the formulation (minimum of the energy stored in the elastic system). It is subsequently shown how the variational form can be discretized into a system of linear ordinary differential equations (more information cane be found in the appendix). Two important time integration techniques that can be used to solve those ODEs are presented in section 3.4. Finally, a very efficient FE formulation based on quadratic tetrahedral elements for real-time applications is detailed.

3.2. Variational formulation

3.2.1. Weak solution in Sobolev spaces

The basic idea of a finite element method is to discretize the physical problem in a so called weak (or variational) form instead of its classical strong formulation. For this purpose we define the test functions $\delta\mathbf{u} \in C_0^\infty \cap V_D(\Gamma_D,0)$ with compact support that vanish on the Dirichlet boundary Γ_D. Then, we demand that the L^2 scalar product

$$\int_{\Omega_0} (\mathrm{Div}\boldsymbol{\sigma} + \rho_0\mathbf{g} - \rho_0\ddot{\mathbf{u}})\,\delta\mathbf{u}\mathrm{dV}_0 = 0 \qquad \forall\delta\mathbf{u} \tag{3.1}$$

between the residual of the partial differential equation (PDE) and the test functions vanishes for all $\delta\mathbf{u}$. Rephrasing the divergence term using the product differentiation rule

$$\mathrm{Div}\boldsymbol{\sigma}\cdot\delta\mathbf{u} = \mathrm{Div}(\boldsymbol{\sigma}\delta\mathbf{u}) - \boldsymbol{\sigma} : \mathrm{Grad}\delta\mathbf{u} \tag{3.2}$$

and application of the divergence theorem yields

$$\int_{\Omega_0} \mathrm{Div}\boldsymbol{\sigma}\cdot\delta\mathbf{u}\mathrm{dV}_0 = \int_{\Omega_0} \mathrm{Div}(\boldsymbol{\sigma}\delta\mathbf{u})\mathrm{dV}_0 - \int_{\Omega_0} \boldsymbol{\sigma} : \mathrm{Grad}\delta\mathbf{u}\mathrm{dV}_0 \tag{3.3}$$

$$= \int_{\Gamma_N} \mathbf{t}\cdot\delta\mathbf{u}\mathrm{dA}_0 - \int_{\Omega_0} \boldsymbol{\sigma} : \mathrm{Grad}\delta\mathbf{u}\mathrm{dV}_0. \tag{3.4}$$

Here, $\mathbf{t}$ are the prescribed traction boundary conditions on Γ_N. Inserting this in eq. 3.1 gives rise to the weak form of Cauchy's equation of motion:

$$\int_{\Omega_0} \boldsymbol{\sigma} : \mathrm{Grad}\delta\mathbf{u}\mathrm{dV}_0 - \int_{\Gamma_N} \mathbf{t}\cdot\delta\mathbf{u}\mathrm{dA}_0 - \int_{\Omega_0} (\rho_0\mathbf{g}\cdot\delta\mathbf{u} + \rho_0\ddot{\mathbf{u}}\cdot\delta\mathbf{u})\,\mathrm{dV}_0 = 0. \tag{3.5}$$

It is apparent, that each $\mathbf{u}$ which solves the original boundary value problem (BVP) is also a solution to the weak formulation. It can also be shown that each $\mathbf{u} \in C^2(\Omega)\cap C^1(\overline{\Omega})\cap V_D(\Gamma_D,\overline{\mathbf{u}})\cap V_N(\Gamma_N,\bar{\mathbf{t}})$ for which eq. (3.5) holds is also a solution of the classical PDE [Bra07]. However, if we only assume $\mathbf{u} \in V_D(\Gamma_D,\overline{\mathbf{u}})\cap V_N(\Gamma_N,\bar{\mathbf{t}})$ and not $\mathbf{u} \in C^2(\Omega)\cap C^1(\overline{\Omega})$, then the weak formulation has more solutions than the classical one. Through the definition of weak derivatives the space of all solutions $\mathbf{u} \in V_D(\Gamma_D,\overline{\mathbf{u}})\cap V_N(\Gamma_N,\bar{\mathbf{t}})$ to eq. 3.5 can be identified with the Sobolev space $H^1(\Omega)$ [Bra07]. It should be noted that there are physical problems (e.g. shock problems) which have no classical solutions, but do have a solution in $H^1(\Omega)$.
The solution space is not only a Sobolev space, but also a Hilbert space (hence the notation $H^1(\Omega)$). Thus, the powerful tools of functional analysis open up a natural way of discretizing the weak form by finite element based techniques. Also, the concept of weak solutions in Sobolev spaces is a very useful tool in the analysis of uniqueness and existence of solutions to BVP. In particular, the Lax-Milgram theorem establishes that the bilinear form (3.1) has a unique solution if it is strongly coercive [Bra07].

3.2.2. Static formulation: Minimizing energy functionals

In the following section we will formulate the static problem in terms of minimizing the total energy of $\mathcal{B}$. It will become apparent that this variational formulation is identical to the weak formulation. This physics-based derivation provides an elegant and intuitive access to weak formulations and will be extensively used in this thesis.
The total potential energy of the system can be formally derived from the balance of mechanical energy for linear elasticity eq. (2.104). For this purpose we assume that each particle in $\mathcal{B}$ is at rest at the beginning of the simulation, i.e. $\mathbf{v}(\mathbf{x},t=0)=0$. Omitting the inertia forces (static approximation) and integrating eq. (2.104) over time then yields

$$\Pi(\mathbf{u}) = \int_{\Omega_0} \boldsymbol{\sigma} : \boldsymbol{\epsilon}\mathrm{dV}_0 - \int_{\partial\Omega_0} \mathbf{t}\cdot\mathbf{u}\mathrm{dA}_0 - \int_{\Omega_0} \rho\mathbf{g}\cdot\mathbf{u}\mathrm{dV}_0. \tag{3.6}$$

The solution to the static problem can be interpreted as the configuration that minimizes this energy functional. The principle of stationary potential energy states a necessary condition for a stationary point in $\Pi(\mathbf{u})$: It requires the directional derivative with respect to the displacements $\mathbf{u}$

$$\Pi(\mathbf{u},\delta\mathbf{u}) = D_{\delta\mathbf{u}}\Pi(\mathbf{u}) = \frac{\mathrm{d}}{\mathrm{dh}}\Pi(\mathbf{u}+h\delta\mathbf{u})\Big|_{h=0} = 0 \tag{3.7}$$

to vanish in all directions [Hol00]. Carrying out the variation on the internal elastic energy yields:

$$\begin{aligned}
D_{\delta\mathbf{u}}\Pi(\mathbf{u})_{int} &= \frac{\mathrm{d}}{\mathrm{dh}}\int_{\Omega_0} \boldsymbol{\sigma} : \boldsymbol{\epsilon}\mathrm{dV}_0\Big|_{h=0} && (3.8)\\
&= \frac{\mathrm{d}}{\mathrm{dh}}\int_{\Omega_0} \boldsymbol{\sigma} : \frac{1}{2}((\nabla\mathbf{u}+h\nabla\delta\mathbf{u}) + (\nabla\mathbf{u}+h\nabla\delta\mathbf{u})^T)\mathrm{dV}_0\Big|_{h=0} && (3.9)\\
&= \int_{\Omega_0} \boldsymbol{\sigma} : \frac{1}{2}(\delta\nabla\mathbf{u} + \delta\nabla\mathbf{u}^T)\mathrm{dV}_0\Big|_{h=0} && (3.10)\\
&= \int_{\Omega_0} \boldsymbol{\sigma} : \delta\frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^T)\mathrm{dV}_0 = \int_{\Omega_0} \boldsymbol{\sigma} : \delta\boldsymbol{\epsilon}\mathrm{dV}_0 && (3.11)
\end{aligned}$$

Similarly, the external energies (loads) can be expressed as

$$D_{\delta\mathbf{u}}\Pi(\mathbf{u})_{ext} = \frac{\mathrm{d}}{\mathrm{dh}}\int_{\partial\Omega_0}\mathbf{t}\cdot(\mathbf{u}+h\delta\mathbf{u})\mathrm{dA}_0 + \frac{\mathrm{d}}{\mathrm{dh}}\int_{\Omega_0}\rho\mathbf{g}\cdot(\mathbf{u}+h\delta\mathbf{u})\mathrm{dV}_0\Big|_{h=0} \tag{3.12}$$

$$= \int_{\partial\Omega_0}\mathbf{t}\cdot\delta\mathbf{u}\mathrm{dA}_0 + \int_{\Omega_0}\rho\mathbf{g}\cdot\delta\mathbf{u}\mathrm{dV}_0 \tag{3.13}$$

and thus the complete variational form is

$$D_{\delta\mathbf{u}}\Pi(\mathbf{u}) = D_{\delta\mathbf{u}}\Pi(\mathbf{u})_{int} + D_{\delta\mathbf{u}}\Pi(\mathbf{u})_{ext} = 0 \tag{3.14}$$

$$\Leftrightarrow \qquad \int_{\Omega_0}\boldsymbol{\sigma}:\delta\boldsymbol{\epsilon}\mathrm{dV}_0 = \int_{\partial\Omega_0}\mathbf{t}\cdot\delta\mathbf{u}\mathrm{dA}_0 + \int_{\Omega_0}\rho\mathbf{g}\cdot\delta\mathbf{u}\mathrm{dV}_0. \tag{3.15}$$

We can thus summarize the variational formulation of the static elasticity problem: Find the displacement field $\mathbf{u}\in V = H^1(\Omega)\cap V_D(\Gamma_D,\overline{\mathbf{u}})\cap V_N(\Gamma_N,\overline{\mathbf{t}})$, s.t. eq. (3.15) holds for all $\delta\mathbf{u}\in H^1\cap V_D(\Gamma_D,0)$. It can be quickly shown (symmetry of $\boldsymbol{\sigma}$) that the variational form is equivalent to the weak formulation.

3.2.3. Dynamic formulation: Hamiltonian variational principle

In the dynamic case, the intuitive formulation in terms of minimizing the elastic energy is replaced by the principle of least action [Ibr09]. More formally we now seek the stationary point of the Hamiltonian variational principle.
With the same assumptions that we made in the static case, we can derive the Lagrangian

$$L(\mathbf{u}) = \Pi(\mathbf{u}) - \mathcal{K}(\dot{\mathbf{u}}) \tag{3.16}$$

$$= \int_{\Omega_0}\boldsymbol{\sigma}:\boldsymbol{\epsilon}\mathrm{dV}_0 - \int_{\partial\Omega_0}\mathbf{t}\cdot\mathbf{u}\mathrm{d}\mathbf{A}_0 - \int_{\Omega_0}\rho\mathbf{g}\cdot\mathbf{u}\mathrm{dV}_0 - \int_{\Omega_0}\frac{1}{2}\rho_0\dot{\mathbf{u}}^2\mathrm{dV}_0 \tag{3.17}$$

of the system [HG01]. The solution to the dynamic elasticity problem can then be regarded as a stationary point of the Hamiltonian variational principle

$$D_{\delta\mathbf{u}}H(\mathbf{u}) = \int_0^T L(\mathbf{u})\mathrm{dt} \tag{3.18}$$

We compute the variation of the kinetic part of the Hamiltonian using integration by parts:

$$D_{\delta\mathbf{u}}H_{kin}(\mathbf{u}) = \frac{\mathrm{d}}{\mathrm{dh}}\int_0^T -\int_{\Omega_0}\frac{1}{2}\rho_0(\frac{\mathrm{d}(\mathbf{u}+h\delta\mathbf{u})}{\mathrm{dt}})^2\mathrm{dV}_0\mathrm{dt}\Big|_{h=0} \tag{3.19}$$

$$= -\int_{\Omega_0}\int_0^T\frac{1}{2}\rho_0 2\frac{\mathrm{d}(\mathbf{u}+h\delta\mathbf{u})}{\mathrm{dt}}\frac{\mathrm{d}}{\mathrm{dh}}\frac{\mathrm{d}(\mathbf{u}+h\delta\mathbf{u})}{\mathrm{dt}}\mathrm{dt}\mathrm{dV}_0\Big|_{h=0} \tag{3.20}$$

$$= \int_{\Omega_0}\left(-\left[\rho_0\frac{\mathrm{d}(\mathbf{u}+h\delta\mathbf{u})}{\mathrm{dt}}\delta\mathbf{u}\right]_0^T + \int_0^T\rho_0\frac{\mathrm{d}^2(\mathbf{u}+h\delta\mathbf{u})}{\mathrm{dt}^2}\delta\mathbf{u}\mathrm{dt}\right)\mathrm{dV}_0\Big|_{h=0} \tag{3.21}$$

$$= \int_{\Omega_0}\left(-\rho_0\frac{\mathrm{d}\mathbf{u}}{\mathrm{dt}}\delta\underbrace{\mathbf{u}(T)}_{=0} + \rho_0\frac{\mathrm{d}\mathbf{u}}{\mathrm{dt}}\delta\underbrace{\mathbf{u}(0)}_{=0} + \int_0^T\rho_0\frac{\mathrm{d}^2\mathbf{u}}{\mathrm{dt}^2}\delta\mathbf{u}\mathrm{dt}\right)\mathrm{dV}_0 \tag{3.22}$$

$$= \int_0^T\int_{\Omega_0}\rho_0\ddot{\mathbf{u}}\delta\mathbf{u}\mathrm{dV}_0\mathrm{dt} \tag{3.23}$$

The final result was obtained by imposing that the variations are zero at both limits of the time interval. By using the previous results obtained in the static case eq. (3.15), the complete variation of the Hamiltonian reads

$$D_{\delta\mathbf{u}}H(\mathbf{u}) = \tag{3.24}$$

$$\int_0^T \left(\int_{\Omega_0} \rho_0 \ddot{\mathbf{u}} \delta\mathbf{u} \mathrm{dV}_0 + \boldsymbol{\sigma} : \delta\boldsymbol{\epsilon} \mathrm{dV}_0 - \int_{\partial\Omega_0} \mathbf{t} \cdot \delta\mathbf{u} \mathrm{dA}_0 - \int_{\Omega_0} \rho\mathbf{g} \cdot \delta\mathbf{u} \mathrm{dV}_0 \right) \mathrm{dt}. \tag{3.25}$$

It is reasonable to require the above equation to hold for all T during the simulation. Thus, the dynamic problem can be posed in its corresponding time-differential, space variational form: Find the displacement field $\mathbf{u} \in V = H^1(\Omega) \cap V_D(\Gamma_D, \overline{\mathbf{u}}) \cap V_N(\Gamma_N, \overline{\mathbf{t}})$, s.t.

$$\int_{\Omega_0} \rho_0 \ddot{\mathbf{u}} \delta\mathbf{u} \mathrm{dV}_0 + \int_{\Omega_0} \boldsymbol{\sigma} : \delta\boldsymbol{\epsilon} \mathrm{dV}_0 - \int_{\partial\Omega_0} \mathbf{t} \cdot \delta\mathbf{u} \mathrm{dA}_0 - \int_{\Omega_0} \rho\mathbf{g} \cdot \delta\mathbf{u} \mathrm{dV}_0 = 0 \tag{3.26}$$

holds for all $\delta\mathbf{u} \in H^1 \cap V_D(\Gamma_D, 0)$.

3.3. Finite element discretization

The idea of the finite element discretization technique is to not look for a solution $\mathbf{u}$ to the variational problem in the infinite-dimensional space V of continuous functions, but to an approximated solution $\mathbf{u}_h$ in a suitable finite-dimensional, discrete sub-space $V_h \subset V$. This subspace is typically spanned by piecewise continuous polynomial functions that have *small* support. In order to define these function spaces, Ω is divided into different polyhedral elements (Fig. 3.1). In this context the boundary of Ω is required to be sufficiently smoothed which is formalized through the assumption that Ω is a Lipschitz domain.

When constructing the finite element space we are facing the fundamental problem that is associated with every discretization technique: How to find the best approximation to the correct solution with a given amount of complexity in terms of degrees of freedom of the discrete system. An important reason for the success of the FE-method are the powerful and elegant tools that are available for a-priori error approximation. At the core, Céa's lemma states that the finite element approximation $\mathbf{u}_h$ is the near best fit to the solution $\mathbf{u}$ in the norm associated with H^1 [Bra07]. Geometrically speaking, the discrete solution $\mathbf{u}_h$ is the orthogonal projection of $\mathbf{u}$ into V_h with respect to the inner product that is induced by the bilinear form (3.1). It follows that the construction of the subspace V_h is crucial for the accuracy of the method.

The convergence of FE methods is typically studied in suitable mesh-dependent norms. Thus, several a-priori error estimates exist for different norms and different finite element types. For finite element spaces that are spanned by piecewise polynomials of order p on tetrahedral and hexahedral elements, it can be shown that the convergence rate for is of order $p+1$ if the real solution is sufficiently smooth. However, the latter is usually not the case in many applications. Thus first or second order polynomials are typically used for elasticity problems.

A finite element mesh $\mathcal{M}$ with elements τ_I and n vertices ν_I is defined to support the basis functions that span the solution space V_h. We denote the set of all elements τ_J around the vertex ν_I with $\mathcal{E}_I$. Similarly $\mathcal{V}_I$ is the set of vertices that form the element τ_I. Typically so called nodal basis functions N_J are used. Each nodal functions N_J satisfies

$$N_J(\nu_I) = \delta_{IJ} \tag{3.27}$$

i.e. it is zero at every other node except the associated J-th node (δ-property). Furthermore, the nodal basis function satisfy the *partition of unity*

$$\sum_{J \in \mathcal{V}_I} N_J(\mathbf{x}) = 1 \quad \forall \mathbf{x} \in \tau_I \tag{3.28}$$

and have a *small*, compact support, i.e.

$$N_J(\mathbf{x}) = 0 \quad \forall \mathbf{x} \notin \mathcal{E}_J. \tag{3.29}$$

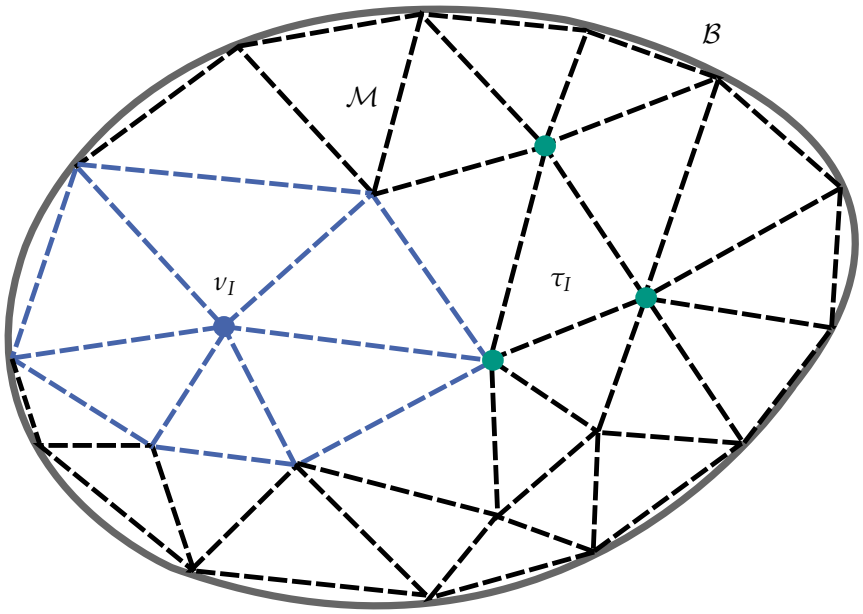


Figure 3.1.: A two dimensional body $\mathcal{B}$ (grey line) is discretized with a triangular finite element mesh $\mathcal{M}$ (dashed lines). The set of vertices $\mathcal{V}_I$ that form the element τ_I is shown in green, while the set of all elements $\mathcal{E}_I$ around the vertex ν_I is colored blue.

The nodal basis functions for linear and quadratic tetrahedral elements can be found in appendix A.4. In order to discretize the weak formulation, $\mathbf{u}_h$ is expressed by the linear combination

$$\mathbf{u}_h(t) = \mathbf{u}_h = \sum_{J=1}^{n} \mathcal{U}_J N_J = \mathcal{U}_J N_J = \mathcal{U}_{jJ} N_{jJ} \tag{3.30}$$

of the n basis functions N_J with the time dependent coefficients $\mathcal{U}$ or $\mathcal{U}(t)$ (in the above equation and from here on out summation over repeated indices is implied). As a direct consequence of the δ-property, the coefficients $\mathcal{U}(t)$ coincide with the displacements of the element nodes when using nodal basis functions. Due to the special form of the basis functions, the solution $\mathbf{u}_h$ is C^0-continuous and we have $V_h \subset V$. It is only through this so called conformal finite element space that statements on uniqueness and existence of solutions can be directly transferred from the continuous to the discrete problem.
For later use, we also introduce the displacement gradient

$$\nabla \mathbf{u} = \nabla(\mathcal{U}_J N_J) = \mathcal{U}_J \nabla N_J. \tag{3.31}$$

It is theoretically possible to use different function spaces for $\mathbf{u}_h$ and the test functions $\delta\mathbf{u}_h$. However, in practice we usually choose these spaces to be the same (Galerkin method). Thus we have

$$\delta\mathbf{u} = \sum_{I=1}^{n} \delta\mathcal{U}_I N_I = \delta\mathcal{U}_I N_I \tag{3.32}$$

3.3.1. Matrix formulation

By inserting the basis functions and the test functions into the variational formulation (3.26) and performing numerical integration, a linear system of ordinary differential equations can be derived. In the following we sketch this procedure and derive the load vector as well as the mass and the stiffness matrix.

We start by inserting the test functions into the expression for the external forces eq. (3.13) to derive

$$D_{\delta\mathcal{U}}\Pi(\mathbf{u}_h)_{ext} = \int_{\partial\Omega_0} \mathbf{t}\cdot(\delta\mathcal{U}_I N_I)\,\mathrm{dA}_0 + \int_{\Omega_0}\rho g\cdot(\delta\mathcal{U}_I N_I)\,\mathrm{dV}_0 \tag{3.33}$$

$$= \delta\mathcal{U}_I\left(\int_{\partial\Omega_0}\mathbf{t}\cdot N_I\mathrm{dA}_0 + \int_{\Omega_0}\rho g\cdot N_I\mathrm{dV}_0\right) = \delta\mathcal{U}_I\mathbf{f}_I^{ext}. \tag{3.34}$$

As the test functions are polynomials, the surface and volume integrals in (3.34) can be accurately and efficiently evaluated using numerical integration techniques such as Gaussian quadrature. For an overview on this topic we refer to Zienkiewicz et al. [ZT77]. The resulting external load vector $\mathbf{f}_I^{ext}$ (sometimes also written as $\mathbf{f}^{ext}$ or f_{iI}^{ext}) denotes the force that acts on each node P_I in the spatial direction j. Its length is thus $3n$.
Similarly, the discretization of the inertia forces (variation of kinetic energy) (3.23) yields

$$D_{\delta\mathcal{U}}\Pi(\mathbf{u}_h)_{kin} = \int_{\Omega_0}\rho\left(\ddot{\mathcal{U}}_J N_J\right)(\delta\mathcal{U}_I N_I)\,\mathrm{dV}_0 = \delta\mathcal{U}_I\int_{\Omega_0}\rho N_J\ddot{\mathcal{U}}_J N_I\mathrm{dV}_0 \tag{3.35}$$

$$= \delta\mathcal{U}_I\left(\int_{\Omega_0}\rho N_J N_I\mathrm{dV}_0\right)\ddot{\mathcal{U}}_J = \delta\mathcal{U}_I\mathbf{M}_{IJ}\ddot{\mathcal{U}}_J. \tag{3.36}$$

The entries of the so called mass matrix $\mathbf{M}_{IJ}$ (sometimes also denoted with $\mathbf{M}$ or $\mathbf{M}_{iIjJ}$) are again computed using numerical integration. The vector $\ddot{\mathcal{U}}_J$ (or simply $\ddot{\mathcal{U}}$) is the vector of nodal accelerations.
The internal forces obviously depend on the deformation field $\mathbf{u}_h$ and thus on the vector of nodal displacement $\mathcal{U}$ (i.e. the coefficients of the basis functions). If the displacement is known we can compute the internal nodal forces using numerical integration. Due to the symmetry of $\boldsymbol{\sigma}$ we can derive:

$$D_{\delta\mathcal{U}}\Pi(\mathbf{u}_h)_{int} = \int_{\Omega_0}\boldsymbol{\sigma}:\delta\boldsymbol{\epsilon}\mathrm{dV}_0 = \int_{\Omega_0}\boldsymbol{\sigma}:\frac{1}{2}\left(\delta\nabla\mathbf{u} + (\delta\nabla\mathbf{u})^T\right)\mathrm{dV}_0 \tag{3.37}$$

$$= \int_{\Omega_0}\boldsymbol{\sigma}:\delta\nabla\mathbf{u}\mathrm{dV}_0 = \int_{\Omega_0}\boldsymbol{\sigma}:(\delta\mathcal{U}_I\nabla N_I)\,\mathrm{dV}_0 \tag{3.38}$$

$$= \delta\mathcal{U}_{iI}\int_{\Omega_0}\sigma_{ik}(\nabla N)_{kI}\mathrm{dV}_0 = \delta\mathcal{U}_{iI}\int_{\Omega_0}\hat{f}_{iI}^{X,int}\mathrm{dV}_0 \tag{3.39}$$

$$= \delta\mathcal{U}_{iI}f_{iI}^{int} = \delta\mathcal{U}_I\mathbf{f}_I^{int} \tag{3.40}$$

Here, $\hat{f}_{iI}^{X,int}$ denotes internal force density. For implicit integration schemes or non-linear static solvers of Newton-Raphson type, the tangent stiffness matrix

$$\mathbf{K}_{IJ} = K_{iIjJ} = \frac{\partial f_{iI}^{int}}{\partial\mathcal{U}_{jJ}} = \int_{\Omega_0}\frac{\partial\hat{f}_{iI}^{X,int}}{\partial\mathcal{U}_{jJ}}\mathrm{dV}_0 = \sum_\tau\int_\tau\frac{\partial\hat{f}_{iI}^{X,int}}{\partial\mathcal{U}_{jJ}}\mathrm{dV}_0 = \sum_\tau\mathbf{K}^\tau \tag{3.41}$$

is necessary. From the above equation it is apparent how the global stiffness matrix $\mathbf{K}_{IJ}$ is constructed: First, the integrals in the above equation are evaluated on a per-element basis. Due to the small support of the basis functions, only few integrals (e.g. 12 for the linear tetrahedron) have to be evaluated for each row. These elemental matrices $\mathbf{K}^\tau$ are then added into the global data structure (*matrix assembly*).
For the linear elastic model, the entries of the tangent stiffness matrix can be computed as (please refer to the appendix A.5 for details):

$$K_{iIjJ} = \int_{\Omega_0}\left(\mu\nabla N_{iJ}\nabla N_{jI} + \mu\delta_{IJ}\sum_{l=1}^{3}\nabla N_{il}\nabla N_{jl} + \lambda\nabla N_{iI}\nabla N_{jJ}\right)\mathrm{dV}_0 \tag{3.42}$$

Here, δ_{IJ} again denotes the Kronecker-delta symbol and μ,λ are the material parameters. For linear elasticity, there is

$$\mathbf{f}^{int} = \mathbf{K}_{IJ}\mathcal{U}_J. \tag{3.43}$$

The global matrices $\mathbf{M},\mathbf{K}$ are symmetric, positive definite sparse matrices. Their sparsity pattern depends on the mesh topology and is thus is irregular for unstructured grids.

Using the obtained discretization of the external, inertia and internal forces we can state the discrete variational form

$$D_{\delta\mathbf{u}}\Pi(\mathbf{u})_{kin} + D_{\delta\mathbf{u}}\Pi(\mathbf{u})_{int} = D_{\delta\mathbf{u}}\Pi(\mathbf{u})_{ext} \tag{3.44}$$

$$\Leftrightarrow \qquad \delta\mathcal{U}_I\mathbf{M}_{IJ}\ddot{\mathcal{U}}_J + \delta\mathcal{U}_I\mathbf{f}^{int} = \delta\mathcal{U}_I\mathbf{f}^{ext}_I \tag{3.45}$$

As the equation holds for all variations $\delta\mathcal{U}_I$, the variational form can be stated as a system of second order ordinary differential equations (ODE):

$$\mathbf{M}\ddot{\mathcal{U}} + \mathbf{f}^{int} = \mathbf{f}^{ext} \tag{3.46}$$

This equality also holds for non-linear problems. For the linear elastic formulation, the internal nodal forces depend linearly on the nodal displacements eq. (3.43) and we can express the equilibrium equations through the following system of linear second order ODE's:

$$\mathbf{M}\ddot{\mathcal{U}} + \mathbf{K}\mathcal{U} = \mathbf{f}^{ext} \tag{3.47}$$

Neumann boundary conditions (forces on the surface) are reflected in the formulation through the external forces $\mathbf{f}^{ext}$. In the following section, we will see how Dirichlet boundary conditions (prescribed displacements) can be imposed during the linear system solve once the system of ODEs has been discretized using time integration techniques.

3.4. Time integration

In this section we derive the full discretization, i.e. we discretize the ODEs in time to derive a linear system of equations. The methods are presented with linear elasticity in mind, but easily generalize to non-linear problems.
For viscoelastic models the internal nodal forces also encapsulate the viscoelastic behavior (see section 2.3.3). Consequently, we can define the tangent damping matrix

$$\mathbf{D}_{IJ} = D_{iIjJ} = \frac{\partial f^{int}_{iI}}{\partial \dot{\mathcal{U}}_{jJ}} \tag{3.48}$$

in addition to the tangent stiffness matrix eq. (3.43). Time integration methods can be categorized into explicit and implicit methods. The explicit methods enforce the equilibrium (3.46) only at the beginning of the time step at the current time t, whereas implicit algorithms enforce it at the end of the time step at time $t + \Delta t$ (Δt is the time step size). Explicit methods are very computationally efficient, as their computation only require matrix-vector operations. In contrast, implicit methods require solving a system of equations for every time step. However, explicit methods are only conditionally stable. Especially for stiff systems, they need very small timesteps to remain stable. Thus they are often a poor choice for soft tissue simulations especially for nearly incompressible material models [SM03]. For real-time deformable model problems, implicit methods have emerged as the dominant means for time discretization [BW98].
In the following, we describe the dynamic equilibrium (3.46) at time $t + \Delta t$ under the assumption that the system is in equilibrium at time t. From now on we denote the point in time for each nodal quantity with left superscript (e.g. $t + \Delta t$). The superscript t which indicates the current time is omitted when apparent from the context. We also define

$$\mathrm{d}\mathcal{U} = {}^{t+\Delta t}\mathcal{U} - {}^{t}\mathcal{U} \qquad \text{and} \qquad \mathrm{d}\dot{\mathcal{U}} = {}^{t+\Delta t}\dot{\mathcal{U}} - {}^{t}\dot{\mathcal{U}}. \tag{3.49}$$

Using the tangent stiffness matrix and the tangent damping matrix, we can express the first order Taylor expansion of $\mathbf{f}^{int}$ around t as

$${}^{t+\Delta t}\mathbf{f}^{int} = {}^{t}\mathbf{f}^{int} + \frac{\partial \mathbf{f}^{int}}{\partial \dot{\mathcal{U}}}({}^{t+\Delta t}\dot{\mathcal{U}} - {}^{t}\dot{\mathcal{U}}) + \frac{\partial \mathbf{f}^{int}}{\partial \mathcal{U}}({}^{t+\Delta t}\mathcal{U} - {}^{t}\mathcal{U}) \tag{3.50}$$

$$= {}^{t}\mathbf{f}^{int} + \mathbf{D}\mathrm{d}\dot{\mathcal{U}} + \mathbf{K}\mathrm{d}\mathcal{U}. \tag{3.51}$$

As the external forces (*dead loads*) do not depend on the deformation, i.e.

$$^{t+\Delta t}\mathbf{f}^{ext} = {}^{t}\mathbf{f}^{ext}, \tag{3.52}$$

the equilibrium at time $t + \Delta t$ is

$$\mathbf{M}\ddot{\mathcal{U}} + \mathbf{D}\mathrm{d}\dot{\mathcal{U}} + \mathbf{K}\mathrm{d}\mathcal{U} = \mathbf{f}^{ext} - \mathbf{f}^{int}. \tag{3.53}$$

In the static, non-linear case the above equation reduces to a Newton-Raphson iteration algorithm. It is solved by first calculating $\mathbf{D}, \mathbf{K}$ for $\mathcal{U}$, then solving eq. 3.53 for $\mathrm{d}\mathcal{U}$ and finally performing the update $\mathcal{U} = \mathcal{U} + \mathrm{d}\mathcal{U}$. This procedure is typically repeated until $\mathrm{d}\mathcal{U}$ is below a certain threshold.

3.4.1. Implicit Euler method

The implicit Euler scheme is unconditionally stable and emerged as the de-facto standard for real-time deformable model simulation. However, it has only a convergence order of one. The update equations for the implicit Euler time integration technique are:

$$\begin{aligned} ^{t+\Delta t}\dot{\mathcal{U}} &= {}^{t}\dot{\mathcal{U}} + h(\,^{t+\Delta t}\ddot{\mathcal{U}}) \\ ^{t+\Delta t}\mathcal{U} &= {}^{t}\mathcal{U} + h(\,^{t+\Delta t}\dot{\mathcal{U}}) \end{aligned} \tag{3.54}$$

Upon inserting these update equations into the equilibrium condition eq. (3.53) we obtain the linear system

$$\mathbf{A}\mathrm{d}\dot{\mathcal{U}} = \mathbf{b} \tag{3.55}$$

with the system matrix

$$\mathbf{A} = \mathbf{M} + \Delta t\mathbf{D} + \Delta t^2\mathbf{K} \tag{3.56}$$

and the effective load

$$\mathbf{b} = \mathbf{f}^{ext} - \mathbf{f}^{int} - \Delta t^2\mathbf{K}\dot{\mathcal{U}}. \tag{3.57}$$

Thus, for each time step one linear system of equations has to be solved. Once the system solve has been performed, the new velocities, displacements and accelerations can be obtained through the update equations (3.54).

3.4.2. Newmark method

In some cases it is desirable to achieve a higher accuracy for the time discretization technique. An attractive alternative is offered by the constant-average-acceleration scheme of the β-Newmark method [BLM00]. It's additional computational overhead is negligible compared to the implicit Euler scheme and it is of second order accuracy. The update equations are:

$$\begin{aligned} ^{t+\Delta t}\mathcal{U} &= \frac{4}{\Delta t^2}(\,^{t+\Delta t}\mathrm{d}\mathcal{U}) - \frac{4}{\Delta t}\,{}^{t}\dot{\mathcal{U}} - {}^{t}\ddot{\mathcal{U}} \\ ^{t+\Delta t}\dot{\mathcal{U}} &= {}^{t}\dot{\mathcal{U}} + \frac{\Delta t}{2}\left({}^{t}\ddot{\mathcal{U}} + {}^{t+\Delta t}\ddot{\mathcal{U}}\right) \end{aligned} \tag{3.58}$$

By again defining a system matrix

$$\mathbf{A} = \frac{4}{\Delta t^2}\mathbf{M} + \frac{2}{\Delta t}\mathbf{D} + \mathbf{K} \tag{3.59}$$

and the effective load

$$\mathbf{b} = {}^{t}\mathbf{f}^{ext} - {}^{t}\mathbf{f}^{int} + \mathbf{M}\left(\frac{4}{\Delta t}\,{}^{t}\dot{\mathcal{U}} + {}^{t}\ddot{\mathcal{U}}\right) + \mathbf{D}\left(2\,{}^{t}\dot{\mathcal{U}}\right) \tag{3.60}$$

the equation to be solved for $\mathrm{d}\mathcal{U}$ can be written as

$$\mathbf{A}\mathrm{d}\mathcal{U} = \mathbf{b} \tag{3.61}$$

We note that in contrast to the above formulation for the implicit Euler scheme, the linear system is formulated in terms of displacements. As we will see in the upcoming section, this allows to easily incorporate movement constraints.

3.4.3. Projection based displacement constraints

We have already seen that Neumann boundary conditions are naturally included in the FE formulation through the external forces $\mathbf{f}^{ext}$. In contrast, displacement constraints have to be explicitly handled. Although it is rather difficult to prescribe displacement constraints at arbitrary points in $\mathcal{B}$ [BLM00], displacement constraints at the element nodes can be incorporated in a very computationally efficient way. In the following paragraph we present how to achieve this when the presented Newmark time integration scheme is used.
If the displacement $\overline{\mathcal{U}}_k$ of certain nodes $P_k, k \in \mathcal{S}$ in the set $\mathcal{S}$ is already known, the dimension of the linear system (3.61) is essentially reduced by the size of $\mathcal{S}$. However, in order to conserve matrix symmetry, the size of the linear system is usually not changed. Instead, the displacements are built into the system by a procedure that is called *displacement projection* (see Alg. 1). The core idea is to project the nodal displacements to the given values (i.e. set $\mathcal{U}_k = \overline{\mathcal{U}}_k \quad \forall k \in \mathcal{S}$) and then modify the linear system in such a way that for the result $\mathrm{d}\mathcal{U}_k = 0 \quad \forall k \in \mathcal{S}$.

Algorithm 1 Newmark timestep with projection constraints

Project nodal displacements, i.e. set $\mathcal{U}_k = \overline{\mathcal{U}}_k \; \forall k \in \mathcal{S}$
Compute ${}^t\mathbf{f}^{int}$
Compute effective load $\mathbf{b}$ according to eq. 3.60
Project $\mathbf{b}$, i.e. set $\mathbf{b}_k = 0 \; \forall k \in \mathcal{S}$
Compute system matrix $\mathbf{A}$ according to eq. 3.59
Project $\mathbf{A}$, i.e. set k-th row and k-th column of $\mathbf{A}$ to 0 and $\mathbf{A}_k k = 1 \; \forall k \in \mathcal{S}$
Solve $\mathbf{A}\mathrm{d}\mathcal{U} = \mathbf{b}$
Update displacements and velocities according to eq. 3.58

3.5. Quadratic corotated tetrahedra

Having established the foundations of FE techniques, a computationally efficient soft tissue model based on corotated quadratic tetrahedra will be presented in this section. The formulation of the method as well as its extensive numerical validation for real-time deformable models and non-rigid registration is based on the respective proceedings publications [SRD+11] [SLH+13].

3.5.1. Corotated finite elements

As previously discussed, linear elastic models cannot be used if an object is subjected to large deformations, regardless of the material properties (linear deformation measures are not rotation invariant). However, if a fully non-linear formulation is used in a static setting or in conjunction with implicit time integration techniques, a non-linear system of equations has to be solved for each time step (see eq. 3.53). Corotated finite elements offer an attractive alternative to this computationally expensive approach and have become a popular choice for real-time deformable models in the realm of computer graphics. The core idea is to linearize the equation of motion by performing the polar decomposition (section 2.2.3)

$$\mathbf{F} = \mathbf{R}\mathbf{U} \tag{3.62}$$

of the deformation gradient $\mathbf{F}$ and using the stretch matrix $\mathbf{U}$ as the deformation measure. In this way a rotation-invariant formulation is achieved. Corotated FE are usually formulated in terms of the current nodal positions and the reference nodal positions, which are denoted by

$$\mathcal{X} = {}^t\mathcal{X} \qquad \text{and} \qquad \mathcal{X}_0 = {}^0\mathcal{X} \tag{3.63}$$

Technically speaking, all occurrences of the deformation gradient φ are substituted by the stretch matrix $\mathbf{U}$. In the following section we briefly show how the complete corotated FE approach emerges through

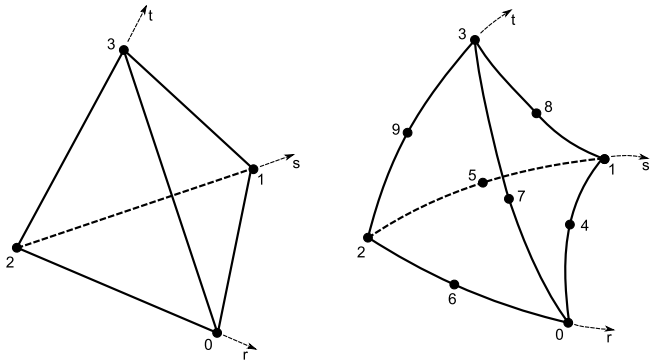


Figure 3.2.: Linear tetrahedron with 4 nodes (tet4) and quadratic tetrahedron with 10 nodes (tet10)

this approach. For a detailed derivation we refer to the appendix A.6. The corotated Cauchy strain tensor can be formulated as

$$\boldsymbol{\epsilon}^{CR} = \frac{1}{2}\left(\mathbf{R}^T \mathcal{X}_J \nabla N_J + (\mathbf{R}^T \mathcal{X}_J \nabla N_J)^T\right) - \mathbf{I} \tag{3.64}$$

The corotated stress $\boldsymbol{\sigma}^{CR}$ is then computed by inserting the corotated strain tensor into the material law eq. (2.99). The corotated nodal forces

$$\mathbf{f}^{int} = \int_{\Omega_0} R_{im} \boldsymbol{\sigma}^{CR}_{mk} (\nabla N)_{kI} \mathrm{dV}_0 \tag{3.65}$$

can be derived accordingly and the corotated tangent stiffness matrix is

$$\mathbf{K}^{CR} = \sum_e \int_\tau \frac{\partial \hat{f}^{X,int}_{iI}}{\partial \mathcal{X}_{jJ}} \mathrm{dV}_0 = \sum_e \int_\tau \mathbf{R} \frac{\partial \hat{\mathbf{f}}^{int}_I}{\partial \mathcal{X}_J} \mathbf{R}^T \mathrm{dV}_0. \tag{3.66}$$

The approach can be described as rotating the deformation field into the initial configuration, calculating the nodal forces using the linear Cauchy strain tensor and finally rotating the forces back to the deformed configuration. By inserting the corotated nodal forces and the corotated stiffness matrix into eq. (3.53), we can finally express the implicit equilibrium:

$$\mathbf{M}\ddot{\mathcal{U}} + \mathbf{D}\mathrm{d}\dot{\mathcal{U}} + \mathbf{K}^{CR}\mathrm{d}\mathcal{U} = \mathbf{f}^{ext} - \mathbf{f}^{int} \tag{3.67}$$

At his point, an important difference to the full non-linear formulation has to be emphasized. When solving for the fully non-linear formulation, a sufficient number of Newton-Raphson steps have to be used in order for the simulation to remain stable. In contrast, the corotated form remains stable when only one Newton-Raphson step is performed each time step. Furthermore, the extraction of the rotational component changes the condition number of the element matrices only marginally which renders the simulation very stable. Although the method cannot model material non-linearities, it offers a very efficient way to achieve a geometrically non-linear formulation. However, in contrast to the linear FEM, the rotation matrices have to be computed and assembled into the stiffness matrix every time step.

3.5.2. Quadratic tetrahedra

The isoparametric 4-node tetrahedron interpolates the position linearly between the nodes (Fig. 3.2). Consequently, the stress is constant over the element and the element shows significant volumetric locking for nearly incompressible materials [BLM00]. In contrast, the 10-node tetrahedron interpolates positions with 2nd order polynomials and doesn't suffer from severe volume locking. Thus, in the realm of mechanical engineering it is well known that quadratic tetrahedra perform much better than linear tetrahedral meshes in many scenarios [CK92]. This is especially true for the simulation of incompressible objects.

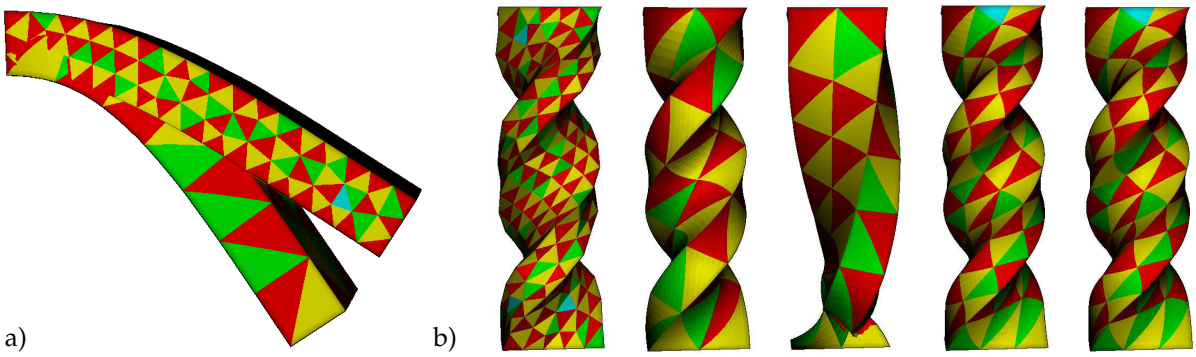


Figure 3.3.: Deformation under gravity (a) and twisting deformation pattern of a beam (b). A 1461 DOF tet4 mesh is compared to a 714 DOF tet10 mesh. The tet10SR element fails to capture the rotation at low resolution (b, middle), but achieves similar accuracy to the tet10 element at higher resolution (b, right).

Typically, the shape functions are defined in a local coordinate system (r,s,t) and all element based operations such as calculating deformations, extracting rotations or numerical integration are performed in this local coordinate system. Polynomial functions (*shape functions*) are used to map the local coordinates to the global coordinate system. If the polynomial degree of the shape functions matches the order of the basis functions, the element is called isoparametric. In case of the isoparametric 10-node tetrahedron, the shape functions are quadratic, which allows curved boundaries and therefore better approximation of the geometry (Fig. 3.2).
As previously mentioned, the integrals that arise during the computation of the forces and the stiffness matrix are evaluated numerically. One cubature point per element is sufficient in order to integrate the stiffness matrix term eq. (3.66) if linear basis functions are used. In contrast, four sample points are necessary to integrate the 10-node tetrahedron. Consequently, four rotation matrices have to be extracted per element for accurate integration. Alternatively, it is possible to extract just one rotation matrix per element by only using the four corner vertices in order to compute the deformation gradient. In this case the element matrix can be pre-computed and the numerical integration can be omitted, which reduces the computational costs by 75%. This simplification will be referred to as the single rotation quadratic tetrahedron.

3.5.3. Numerical validation

Numerical simulation studies on a simple beam geometry are performed in order to compare the efficiency of the linear tetrahedral (tet4), the quadratic tetrahedral (tet10) as well as the single rotation quadratic tetrahedral (tet10SR) elements. In the first scenario, the beam deforms under gravity, while it is subjected to a twisting deformation pattern in the second scenario (Fig. 3.3). Incompressible material models (Poisson's ration $\nu = 0.49$) are used throughout the study. The linear and the quadratic corotated FEM were implemented using the Simulation Open Framework Architecture (SOFA) toolkit [FDD+12]. For all simulations in this study, the Newmark time integration scheme is used along with the Pardiso direct sparse solver from the Intel MKL 10.3.
In order to perform a quantitative analysis of the discretization error for each model, a reference solution on a high resolution quadratic mesh (100k elements) is computed for both problems. Test models of different resolutions are subsequently compared to this reference model. We choose the root mean squared (RMS) error at the nodes of the reference solution as the error measures. The RMS errors with respect to the degrees of freedom (DOF) are depicted in Fig. 3.4.
In case of the gravity induced deformation, tet4 elements need much more DOF (up to 40x) than tet10 elements in order to achieve the same accuracy. This result illustrates the locking behavior of linear tetrahedral elements. It should also be pointed out that the tet10SR formulation shows only negligible difference to the fully integrated tet10 element. Due to the displacement boundary conditions and the

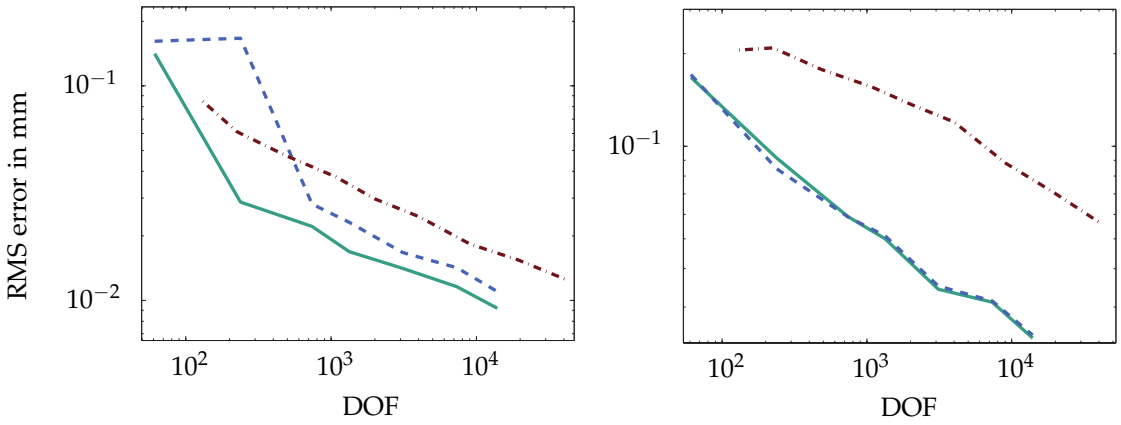


Figure 3.4.: RMS error over DOF for twisting deformation (left) and gravity induced deformation (right). The tet10 (green) and tet10SR (blue) elements show far superior accuracy for the same DOF than the tet4 (red) mesh.

absence of volumetric forces, the difference in the twisting scenario is not as pronounced. However, the tet4 elements still need an order of magnitude more DOFs. It is apparent from Fig. 3.3 that the tet4 mesh with 1461 DOFs still shows visible edges, while the lower resolution tet10 mesh (714 DOFs) produces a smooth deformation. It can also be seen that the tet10SR element fails to capture the rotations at low resolution (Fig. 3.3), but achieves a similar accuracy as the tet10 element for higher resolutions (right). It can be concluded that quadratic elements are superior to tet4 elements in both scenarios. Furthermore, the single rotation quadratic tetrahedral formulation (tet10SR) can be a computationally efficient alternative to the standard tet10 formulation for many applications.

A.

Additional remarks on elasticity theory and FE methods

A.1. Basics of vector analysis

A.1.1. Divergence theorem

For any smooth vector field $\mathbf{v}(\mathbf{x})$ and any smooth tensor field $\mathbf{A}(\mathbf{x})$ that are defined on a compact region Ω which is enclosed by the smooth, closed surface $\partial\Omega$ we have

$$\int_{\partial\Omega} \mathbf{v}\cdot\mathbf{n}\mathrm{dA} = \int_{\Omega} \mathrm{div}\mathbf{v}\mathrm{dV} \qquad \text{or} \qquad \int_{\partial\Omega} v_i n_i \mathrm{dA} = \int_{\Omega} \frac{\partial v_i}{\partial x_i}\mathrm{dV} \tag{A.1}$$

$$\int_{\partial\Omega} \mathbf{A}\mathbf{n}\mathrm{dA} = \int_{\Omega} \mathrm{div}\mathbf{A}\mathrm{dV} \qquad \text{or} \qquad \int_{\partial\Omega} A_{ij} n_j \mathrm{dA} = \int_{\Omega} \frac{\partial A_{ij}}{\partial x_j}\mathrm{dV} \tag{A.2}$$

A.1.2. Product differentiation rules for tensors

For any smooth vector field $\mathbf{v}(\mathbf{x})$ and any smooth tensor field $\mathbf{A}(\mathbf{x})$ we have

$$\mathrm{div}(\mathbf{A}^T\mathbf{v}) = \mathrm{div}(\mathbf{A})\cdot\mathbf{v} + \mathrm{div}(\mathbf{A}) : \mathrm{grad}(\mathbf{v}) \tag{A.3}$$

A.2. Work conjugancy of S and $\dot{\mathbf{E}}$

It has already been shown in section (2.3.1) that the first Piola-Kirchhoff stress tensor $\mathbf{P}$ is work conjugated to the material time derivative of the deformation gradient $\mathbf{F}$. It is quickly shown that the material time derivative of the Cauchy-Green strain tensor $\mathbf{C}$

$$\dot{\mathbf{C}} = \frac{\mathrm{D}}{\mathrm{Dt}}\left(\mathbf{F}^T\mathbf{F}\right) = \dot{\mathbf{F}}^T\mathbf{F} + \mathbf{F}^T\dot{\mathbf{F}} \tag{A.4}$$

related to the material time derivative of the Green-Lagrange strain tensor by

$$\dot{\mathbf{E}} = \frac{\mathrm{D}}{\mathrm{Dt}}\frac{1}{2}\left(\mathbf{F}^T\mathbf{F} - \mathbf{I}\right) = \frac{1}{2}\left(\dot{\mathbf{F}}^T\mathbf{F} + \mathbf{F}^T\dot{\mathbf{F}}\right) = \frac{1}{2}\dot{\mathbf{C}}. \tag{A.5}$$

Based on the results of section 2.3.1 (work conjugancy of $\mathbf{P}$ and $\dot{\mathbf{F}}$), we can thus derive

$$\mathbf{P} : \dot{\mathbf{F}} = \mathrm{tr}(\mathbf{P}^T\dot{\mathbf{F}}) = \mathrm{tr}(\dot{\mathbf{F}}^T\mathbf{P}) = \mathrm{tr}(\dot{\mathbf{F}}^T\mathbf{F}\underbrace{\mathbf{F}^{-1}\mathbf{P}}_{\mathbf{S}}) \tag{A.6}$$

$$= \mathrm{tr}(\mathbf{S}\dot{\mathbf{F}}^T\mathbf{F}) = \mathrm{tr}(\mathbf{S}\mathbf{F}\dot{\mathbf{F}}^T) = \mathrm{tr}(\mathbf{S}\frac{1}{2}(\dot{\mathbf{F}}^T\mathbf{F} + \mathbf{F}\dot{\mathbf{F}}^T)) \tag{A.7}$$

$$= \frac{1}{2}\mathbf{S} : \dot{\mathbf{C}} = \mathbf{S} : \dot{\mathbf{E}} \tag{A.8}$$

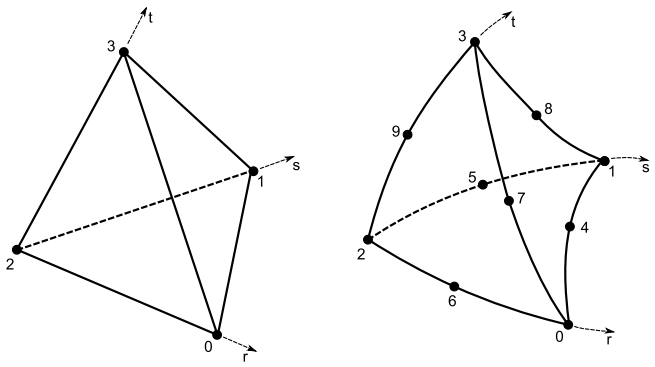


Figure A.1.: Linear tetrahedron with 4 nodes (tet4) and quadratic tetrahedron with 10 nodes (tet10)

A.3. The Saint Venant-Kirchhoff model

The elastic energy functional for Saint-Venant Kirchhoff model is defined by

$$\Psi(\mathbf{E}) = \frac{\lambda}{2}(\mathrm{tr}\mathbf{E})^2 + \mu \mathrm{tr}(\mathbf{E}^2). \tag{A.9}$$

In order to express the relation

$$\mathbf{S}(\mathbf{E}) = \frac{\partial \Psi(\mathbf{E})}{\partial \mathbf{E}}, \tag{A.10}$$

we note the partial derivatives

$$\frac{\partial \mathrm{tr}(\mathbf{E})}{\partial \mathbf{E}} = \frac{1}{\partial E_{ij}} \sum_{k=1}^{3} E_{kk} = \delta_{ij} = \mathbf{I} \tag{A.11}$$

and

$$\mathbf{S}(\partial \mathbf{E}) = \frac{\partial \mathrm{tr}(\mathbf{E}^2)}{\partial \mathbf{E}} = \frac{1}{\partial E_{ij}} \sum_{k,l=1}^{3} E_{kl} E_{kl} = 2E_{ij} = 2\mathbf{E}. \tag{A.12}$$

Thus, we obtain

$$\frac{\partial \Psi(\mathbf{E})}{\partial \mathbf{E}} = \frac{\lambda}{2} \frac{\partial \left((\mathrm{tr}\mathbf{E})^2\right)}{\partial \mathbf{E}} + \mu \frac{\partial \left(\mathrm{tr}(\mathbf{E}^2)\right)}{\partial \mathbf{E}} = \lambda \mathrm{tr}\mathbf{E}\mathbf{I} + 2\mu\mathbf{E}. \tag{A.13}$$

A.4. Polynomial shape functions for selected elements

A.4.1. Nodal shape functions in local (curvilinear) coordinates

Linear 4-node tetrahedron (node numbering as shown in Fig. A.1):

$$N_0(r,s,t) = r \tag{A.14}$$
$$N_2(r,s,t) = s \tag{A.15}$$
$$N_3(r,s,t) = 1 - r - s - t \tag{A.16}$$
$$N_4(r,s,t) = t \tag{A.17}$$
$$\tag{A.18}$$

Quadratic 10-node tetrahedron (node numbering as shown in Fig. A.1):

$$N_0(r,s,t) = (2r-1)r \tag{A.19}$$
$$N_1(r,s,t) = (2s-1)s \tag{A.20}$$
$$N_2(r,s,t) = (2(1-r-s-t)-1)(1-r-s-t) \tag{A.21}$$
$$N_3(r,s,t) = (2t-1)t \tag{A.22}$$
$$N_4(r,s,t) = 4rs \tag{A.23}$$
$$N_5(r,s,t) = 4(1-r-s-t)s \tag{A.24}$$
$$N_6(r,s,t) = 4(1-r-s-t)r \tag{A.25}$$
$$N_7(r,s,t) = 4st \tag{A.26}$$
$$N_8(r,s,t) = 4rt \tag{A.27}$$
$$N_9(r,s,t) = 4(1-r-s-t)t \tag{A.28}$$
$$\tag{A.29}$$

A.4.2. Nodal shape functions in global coordinates

The standard triangle [(1,0,0) (0,1,0) (0,0,0) (0,0,1)] is mapped to the global coordinate system using the shape functions (isometric elements). If $\mathbf{X}_k$ denotes the initial position of the k-th node of the n-node tetrahedron, we define the function $\xi(\mathbf{r})$ that maps the local coordinates $\mathbf{r} = (r,s,t)$ to the global coordinates $\mathbf{X} = (X_1, X_2, X_3)$:

$$\xi(\mathbf{r}) = \mathbf{X}(\mathbf{r}) = \mathbf{X}(r,s,t) = \sum_{k=0}^{n} \mathbf{X}_k N_k(r,s,t) \tag{A.30}$$

The global derivatives of the shape functions are

$$\frac{\partial N_I(\mathbf{x})}{\partial \mathbf{X}} = \mathbf{J}^{-1} \frac{\partial N_I(\mathbf{x})}{\partial \mathbf{r}} \tag{A.31}$$

where $\mathbf{J}$ is the jacobian

$$\mathbf{J} = \nabla\xi = \begin{pmatrix} \frac{\partial X_1}{\partial r} & \frac{\partial X_2}{\partial r} & \frac{\partial X_3}{\partial r} \\ \frac{\partial X_1}{\partial s} & \frac{\partial X_2}{\partial s} & \frac{\partial X_3}{\partial s} \\ \frac{\partial X_1}{\partial t} & \frac{\partial X_2}{\partial t} & \frac{\partial X_3}{\partial t} \end{pmatrix} \tag{A.32}$$

A.5. Internal nodal forces and the stiffness matrix for linear elasticity

As detailed in section 3.3.1, the internal nodal forces are given by

$$\mathbf{f}^{int} = f_{iI}^{int} = \int_{\Omega_0} \hat{f}_{iI}^{X,int} \mathrm{dV}_0 = \sum_e \int_\tau \hat{f}_{iI}^{X,int} \mathrm{dV}_0 = \int_{\Omega_0} \sigma_{ik} (\nabla N)_{kI} \mathrm{dV}_0. \tag{A.33}$$

By noting that the gradient of the displacement vector reads

$$\nabla \mathbf{u} = \nabla(\mathcal{U}_I N_I) = \mathcal{U}_I \nabla N_I = \mathcal{U}_{iI} (\nabla N)_{kI}, \tag{A.34}$$

we can express its the derivative with respect to the nodal displacements through

$$\frac{\partial}{\partial \mathcal{U}_{jJ}} (\nabla \mathcal{U})_{ik} = \frac{\partial}{\partial \mathcal{U}_{jJ}} (\mathcal{U}_{iI} (\nabla N)_{kI}) = \delta_{ij} (\nabla N_{kJ}). \tag{A.35}$$

The derivative of the Cauchy strain with respect to the nodal displacements can thus be derived as

$$\begin{aligned}\frac{\partial \epsilon_{ik}}{\partial \mathcal{U}_{jJ}} &= \frac{1}{2}\frac{\partial}{\partial \mathcal{U}_{jJ}}\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) \\ &= \frac{1}{2}\frac{\partial}{\partial \mathcal{U}_{jJ}}\left(\mathcal{U}_{iI}(\nabla N)_{kI} + \mathcal{U}_{kI}(\nabla N)_{iI}\right) \\ &= \frac{1}{2}\left(\delta_{ij}(\nabla N)_{kJ} + \delta_{kj}(\nabla N)_{iJ}\right). \end{aligned} \tag{A.36}$$

Consequently, the derivation of the Cauchy stress follows as

$$\begin{aligned}\frac{\partial \sigma_{ik}}{\partial \mathcal{U}_{jJ}} &= 2\mu\frac{\partial \epsilon_{ik}}{\partial \mathcal{U}_{jJ}} + \lambda\delta_{ik}\frac{\partial \epsilon_{ll}}{\partial \mathcal{U}_{jJ}} \\ &= \mu\left(\delta_{ij}(\nabla N)_{kJ} + \delta_{kj}(\nabla N)_{iJ}\right) + \frac{\lambda}{2}\delta_{ik}\left(\delta_{lj}(\nabla N)_{lJ} + \delta_{lj}(\nabla N)_{lJ}\right) \\ &= \mu\left(\delta_{ij}(\nabla N)_{kJ} + \delta_{kj}(\nabla N)_{iJ}\right) + \frac{\lambda}{2}2\delta_{ik}(\nabla N)_{jJ} \\ &= \mu\delta_{ij}(\nabla N)_{kJ} + \mu\delta_{kj}(\nabla N)_{iJ} + \lambda\delta_{ik}(\nabla N)_{jJ} \end{aligned} \tag{A.37}$$

and the derivative of the internal force density can be expressed as

$$\begin{aligned}\frac{\partial \hat{f}^{X,int}_{iI}}{\partial \mathcal{U}_{jJ}} &= \frac{\partial}{\partial \mathcal{U}_{jJ}}\left(\sigma_{ik}(\nabla N)_{kI}\right)\mathrm{dV}_0 \\ &= \frac{\partial \sigma_{ik}}{\partial \mathcal{U}_{jJ}}(\nabla N)_{kI} \\ &= \left(\mu\delta_{ij}(\nabla N)_{kJ} + \mu\delta_{kj}(\nabla N)_{iJ} + \lambda\delta_{ik}(\nabla N)_{jJ}\right)(\nabla N_I)_{kI} \\ &= \left(\mu\delta_{ij}(\nabla N)_{kJ} + \mu\delta_{kj}(\nabla N)_{iJ} + \lambda\delta_{ik}(\nabla N)_{jJ}\right)(\nabla N_I)_{kI} \\ &= \mu\delta_{ij}(\nabla N)_{kJ}(\nabla N_I)_{kI} + \mu(\nabla N)_{iJ}(\nabla N_I)_{jI} \\ &\quad +\lambda(\nabla N)_{jJ}(\nabla N_I)_{iI}. \end{aligned} \tag{A.38}$$

This yields the stiffness matrix

$$\frac{\partial f^{int}_{iI}}{\partial \mathcal{U}_{jJ}} = \int_{\Omega_0}\frac{\partial \hat{f}^{X,int}_{iI}}{\partial \mathcal{U}_{jJ}}\mathrm{dV}_0 = \sum_e\int_\tau \frac{\partial \hat{f}^{X,int}_{iI}}{\partial \mathcal{U}_{jJ}}\mathrm{dV}_0 = \sum_e \mathbf{K}^e = \mathbf{K}. \tag{A.39}$$

A.6. Internal nodal forces and the stiffness matrix for corotated elasticity

The polar decomposition of deformation gradient (see section 2.2.3)

$$\mathbf{F} = \mathbf{R}\mathbf{U} \tag{A.40}$$

is used to define the stretch measure

$$\mathbf{U} = \mathbf{R}^T\mathbf{F}. \tag{A.41}$$

We express the displacement gradient through the nodal positions

$$\nabla \mathbf{u} = \mathbf{F} - \mathbf{I} = \nabla(\mathcal{X}_J N_J) = \mathcal{X}_J \nabla N_J - \mathbf{I} \tag{A.42}$$

in order to relate the Cauchy strain to the nodal positions:

$$\begin{aligned}\boldsymbol{\epsilon} &= \frac{1}{2}\left((\mathbf{F}-\mathbf{I})+(\mathbf{F}-\mathbf{I})^T\right)=\frac{1}{2}\left((\mathcal{X}_J\nabla N_J-\mathbf{I})+(\mathcal{X}_J\nabla N_J-\mathbf{I})^T\right)\\ &= \frac{1}{2}\left(\mathcal{X}_{iJ}(\nabla N)_{jJ}+\mathcal{X}_{jJ}(\nabla N)_{iJ}\right)-\mathbf{I}. \end{aligned} \tag{A.43}$$

For corotated elasticity, the deformation gradient is replaced by the stretch measure

$$\mathbf{U}=\mathbf{R}^T\nabla\mathbf{F}=R_{ki}\mathcal{X}_{kJ}(\nabla N)_{jJ}. \tag{A.44}$$

Consequently, the corotated Cauchy strain reads

$$\boldsymbol{\epsilon}^{CR} = \boldsymbol{\epsilon}^{CR}_{ij}=\frac{1}{2}\left(\mathbf{R}^T\mathcal{X}_J\nabla N_J+(\mathbf{R}^T\mathcal{X}_J\nabla N_J)^T\right)-\mathbf{I} \tag{A.45}$$

and the corotated stress is derived to

$$\boldsymbol{\sigma}^{CR}_{ij} = 2\mu\boldsymbol{\epsilon}^{CR}_{ij}+\lambda\delta_{ij}\boldsymbol{\epsilon}^{CR}_{ll}.$$

We note that the internal nodal forces can be written in terms of the variation of nodal position

$$\begin{aligned} D_{\delta\mathcal{U}}\Pi(\mathbf{u}_h)_{int} &= \int_{\Omega_0}\boldsymbol{\sigma}:\delta\boldsymbol{\epsilon}\mathrm{dV}_0=\int_{\Omega_0}\boldsymbol{\sigma}:\frac{1}{2}\left(\delta\nabla\mathbf{u}+(\delta\nabla\mathbf{u})^T\right)\mathrm{dV}_0\\ &= \int_{\Omega_0}\boldsymbol{\sigma}:\delta\nabla\mathbf{u}\mathrm{dV}_0=\int_{\Omega_0}\boldsymbol{\sigma}:(\delta\mathcal{U}_I\nabla N_I)\,\mathrm{dV}_0\\ &= \int_{\Omega_0}\boldsymbol{\sigma}:(\delta\mathcal{X}_I\nabla N_I)\,\mathrm{dV}_0 \end{aligned} \tag{A.46}$$

in order to derive the corotated nodal forces

$$\begin{aligned} D_{\delta\mathcal{U}}\Pi(\mathbf{u}_h)_{int} &= \int_{\Omega_0}\boldsymbol{\sigma}:\left(\mathbf{R}^T\delta\mathcal{X}_I\nabla N_I\right)\mathrm{dV}_0=\int_{\Omega_0}(\mathbf{R}\boldsymbol{\sigma}):(\delta\mathcal{X}_I\nabla N_I)\,\mathrm{dV}_0\\ &= \delta\mathcal{X}_{iI}\int_{\Omega_0}\mathbf{R}_{im}\boldsymbol{\sigma}_{mk}(\nabla N)_{kI}\mathrm{dV}_0=\delta\mathcal{X}_{iI}\int_{\Omega_0}\hat{f}^{int,CR}_{iI}\mathrm{dV}_0\\ &= \delta\mathcal{X}_{iI}f^{int,CR}_{iI}=\delta\mathcal{X}_I\mathbf{f}^{int,CR}_I \end{aligned} \tag{A.47}$$

with

$$\hat{f}^{int,CR}_{iI}=\int_{\Omega_0}\mathbf{R}_{im}\boldsymbol{\sigma}^{CR}_{mk}(\nabla N)_{kI}\mathrm{dV}_0. \tag{A.48}$$

In order to formulate the corotated stiffness matrix, we first compute the derivate of the corotated Cauchy strain with respect to the nodal positions:

$$\begin{aligned}\frac{\partial\boldsymbol{\epsilon}_{mk}}{\partial\mathcal{X}_{jJ}} &= \frac{1}{2}\frac{\partial}{\partial\mathcal{X}_{jJ}}\left(R_{nm}F_{nk}+F_{nm}R_{nk}\right)-\mathbf{I}\\ &= \frac{1}{2}\frac{\partial}{\partial\mathcal{X}_{jJ}}\left(R_{nm}\mathcal{X}_{nI}(\nabla N)_{kI}+\mathcal{X}_{nI}(\nabla N)_{mI}R_{nk}\right)\\ &= \frac{1}{2}\left(\delta_{nj}R_{nm}(\nabla N)_{kJ}+\delta_{nj}(\nabla N)_{mJ}R_{nk}\right)\end{aligned} \tag{A.49}$$

In a similar fashion as in the linear elastic case we derive

$$\begin{aligned}&\frac{\partial\boldsymbol{\sigma}^{CR}_{mk}}{\partial\mathcal{X}_{jJ}}=2\mu\frac{\partial\boldsymbol{\epsilon}^{CR}_{mk}}{\partial\mathcal{X}_{jJ}}+\lambda\delta_{mk}\frac{\partial\boldsymbol{\epsilon}^{CR}_{ll}}{\partial\mathcal{X}_{jJ}}\\ &=\mu\left(\delta_{nj}R_{nm}(\nabla N)_{kJ}+\delta_{nj}(\nabla N)_{mJ}R_{nk}\right)\\ &+\frac{\lambda}{2}\delta_{mk}\left(\delta_{nj}R_{nl}(\nabla N)_{lJ}+\delta_{nj}(\nabla N)_{lJ}R_{nl}\right)\\ &=\mu\left(\delta_{nj}R_{nm}(\nabla N)_{kJ}+\delta_{nj}(\nabla N)_{mJ}R_{nk}\right)+\lambda\delta_{mk}\delta_{nj}R_{nl}(\nabla N)_{lJ}\end{aligned} \tag{A.50}$$

and express the corotated force density through

$$\begin{aligned}
&\frac{\partial \hat{f}_{iI}^{X,int}}{\partial \mathcal{X}_{jJ}} = \frac{\partial}{\partial \mathcal{X}_{jJ}} \mathbf{R}_{im} \sigma_{mk}^{CR} (\nabla N)_{kI} = \mathbf{R}_{im} \frac{\partial \sigma_{mk}^{CR}}{\partial \mathcal{X}_{jJ}} (\nabla N)_{kI} \\
= \;& \mathbf{R}_{im} \left(\mu \left(\delta_{nj} R_{nm} (\nabla N)_{kJ} + \delta_{nj} (\nabla N)_{mJ} R_{nk} \right) + \lambda \delta_{mk} \delta_{nj} R_{nl} (\nabla N)_{lJ} \right) (\nabla N)_{kI} \\
= \;& \mathbf{R}_{im} \mu \left(\delta_{nm} R_{jn} (\nabla N)_{kJ} (\nabla N)_{kI} + (\nabla N)_{mJ} R_{jk} (\nabla N)_{kI} \right) \\
& + \lambda R_{jl} (\nabla N)_{lJ} (\nabla N)_{mI} \\
= \;& \mathbf{R}_{im} \mu \left(\delta_{nm} (\nabla N)_{kJ} (\nabla N)_{kI} R_{jn} + (\nabla N)_{mJ} (\nabla N)_{nI} \right) R_{jn} \\
& + \lambda (\nabla N)_{nJ} (\nabla N)_{mI} R_{jn} \\
= \;& \mathbf{R}_{im} \left(\mu \left(\delta_{nm} (\nabla N)_{kJ} (\nabla N)_{kI} + (\nabla N)_{mJ} (\nabla N)_{nI} \right) + \lambda (\nabla N)_{nJ} (\nabla N)_{mI} \right) R_{jn} \\
= \;& \mathbf{R}_{im} \frac{\partial \hat{f}_{mI}^{int}}{\partial \mathcal{X}_{nJ}} R_{jn} = \mathbf{R} \frac{\partial \mathbf{f}_I^{int,CR}}{\partial \mathcal{X}_J} \mathbf{R}^T.
\end{aligned} \tag{A.51}$$

The global corotated stiffness matrix is then given by

$$\mathbf{K}^{CR} = \sum_e \int_\tau \frac{\partial \hat{f}_{iI}^{X,int}}{\partial \mathcal{X}_{jJ}} \mathrm{dV}_0 = \sum_e \int_\tau \mathbf{R} \frac{\partial \hat{\mathbf{f}}_I^{int}}{\partial \mathcal{X}_J} \mathbf{R}^T \mathrm{dV}_0. \tag{A.52}$$

B.

Glossary

$\mathbf{A}$	arbitrary matrix (e.g. system matrix for linear system)
$\mathcal{A}$	additional DOF for sign enriched elements
$\mathcal{B}$	body (object) under analysis
$\mathbf{C}$	Cauchy-Green strain tensor
$\overline{\mathbf{C}}$	modified (isochoric) Cauchy-Green strain tensor
$\mathbf{D}$	damping matrix
$\mathbf{d}$	rate of deformation tensor
$\mathrm{d}\mathbf{A}, \mathrm{d}A\mathbf{n}$	infinitesimal area element in current configuration
$\mathrm{d}\mathbf{A}_0, \mathrm{d}A_0\mathbf{N}$	infinitesimal area element in reference configuration
div	divergence operator in current configuration
Div	divergence operator in reference configuration
$\mathrm{d}V$	infinitesimal volume element in current configuration
$\mathrm{d}V_0$	infinitesimal volume element in reference configuration
$\mathrm{d}\mathbf{x}, \mathrm{d}x_i$	line element in current configuration
$\mathrm{d}\mathbf{X}, \mathrm{d}X_i$	line element in reference configuration
$\mathrm{d}\mathcal{U}$	difference vector of nodal displacements
$\mathrm{d}\dot{\mathcal{U}}$	difference vector of nodal velocities
$\mathrm{d}\mathcal{X}$	difference vector of nodal positions
Δt	increment during time integration
$\mathbf{E}$	Green-Lagrange strain tensor
$\mathcal{E}$	set of elements in $\mathcal{M}$
$\boldsymbol{\epsilon}$	infinitesimal strain tensor
$\mathbf{f}^{ext}$, $\mathbf{f}_I^{ext}$, f_{iI}^{ext}	vector of external nodal forces
$\mathbf{f}^{int}$, $\mathbf{f}_I^{int}$, f_{iI}^{int}	vector of internal nodal forces
$\hat{f}_{iI}^{X,int}$	internal force density
$\mathbf{f}^{int,CR}$, $\mathbf{f}_I^{int,CR}$, $f_{iI}^{int,CR}$	vector of corotated internal nodal forces
$\hat{f}_{iI}^{int,CR}$	internal force density
$\mathbf{F}$	deformation gradient tensor

$\dot{\mathbf{F}}$	time derivative of the deformation gradient tensor
$\overline{\mathbf{F}}$	modified (isochoric) deformation gradient tensor
Γ_C	boundary of the cut with zero Neumann (force) boundary conditions
Γ_N	boundary with Dirichlet (displacement) boundary conditions
Γ_D	boundary with Neumann (force) boundary conditions
$\mathcal{G}$	Cauchy-elastic response function
grad	gradient operator in current configuration
Grad	gradient operator in reference configuration
$\mathbf{g}$	gravity acceleration vector
$\mathcal{H}$	hyperelastic response function
H	Hamiltonian
$\mathbf{I}$	identity matrix
$\mathbf{l}$	spatial velocity gradient
$\mathbf{K}$	stiffness matrix
$\mathcal{K}$	kinetic energy
L	Lagrangian
$\mathbf{M}$	mass matrix
$\mathcal{M}$	finite element mesh
N_J, N_{jJ}	basis functions
N_I, N_{iI}	test functions
$\mathbf{n}$	normal in current configuration
$\mathbf{N}$	normal in reference configuration
ν	vertex in $\mathcal{S}$
$\mathbf{O}$	lifting operator
Ω_0	region that is occupied by $\mathcal{B}$ in reference configuration
Ω	region that is occupied by $\mathcal{B}$ in current configuration
$\mathbf{P}$	first Piola-Kirchhoff stress tensor
$\mathbf{p}$	point in $\mathcal{B}$
$\mathcal{P}_{ext}$	rate of external mechancial work (power input)
Π	potential energy functional
$\mathcal{P}_{int}$	rate of internal mechanical work
Ψ	internal elastic energy
$\mathbf{R}$	rotation matrix obtained from polar decomposition of $\mathbf{F}$
ρ	densitiy in current configuration
ρ_0	density in reference configuration
$\mathbf{S}$	second Piola-Kirchhoff stress tensor
$\mathcal{S}$	triangular surface mesh

σ	Cauchy stress tensor
$\mathbf{t}$	traction vector in current configuration
$\mathbf{T}$	traction vector in reference configuration
$\mathbb{T}$	element in $\mathcal{S}$
τ	element in $\mathcal{M}$
tr	trace operator
$\mathbf{U}$	stretch matrix obtained from polar decomposition of $\mathbf{F}$
$\mathbf{u}, u_i$	displacement field
$\mathcal{U}, \mathcal{U}_J, \mathcal{U}_{jJ}$	nodal displacement field
$\dot{\mathcal{U}}$	vector of nodal velocities
$\ddot{\mathcal{U}}$	vector of nodal accelerations
$\delta\mathbf{u}$	variation of the displacement field
$\mathbf{w}$	rate of rotation tensor
$\mathbf{v}$	velocity field
$\dot{\mathbf{v}}$	acceleration field
$\mathcal{V}$	set of vertices in $\mathcal{M}$
$V_D(\Gamma_D, \overline{\mathbf{u}})$	space of functions that fulfill the Dirichlet boundary conditions on Γ_D
$V_N(\Gamma_N, \bar{\mathbf{t}})$	space of functions that fulfill the Neumann boundary conditions on Γ_N
w_{int}	stress power per unit reference volume
$\mathbf{x}, x_i$	positional field in the current configuration
$\mathbf{X}, X_i$	positional field in the reference configuration
$\mathcal{X}, \mathcal{X}_J, \mathcal{X}_{jJ}$	nodal positions

Part II.

Tutorials

C.

Getting started

This chapter describes how to work with virtualized environments that contain pre-configured simulation packages. There are several such packages available. Each contains an implementation of the Medical Simulation Markup Language (MSML) as well as different simulation backends. In the following we describe how to run these environments based on Docker containers[1]. We also describe how to connect to server instances that run dockerized MSML environments.

C.1. Running MSML-based Docker containers

In this section, we describe how to run a pre-configured deployment of the Simulation Open Framework Architecture (SOFA) simulation package.

C.1.1. Installing the container

1. Install the Docker framework for your Linux distribution. Further details can be found on the Docker homepage. For Ubuntu Linux there is the possibility to use either pre-configured packages (see `www.ubuntuupdates.org/ppa/docker` or to install Docker manually (`http://docs.docker.com/installation/ubuntulinux/`).
2. Optional: You can add your user to the docker group in order to avoid the use ofsudo in front of the docker commands:

   ```
   $ sudo -aG docker username
   ```

   Please remember to log out in order to for the changes to take effect.
3. Pull the SOFA container from the Docker Hub:

   ```
   $ docker pull ssuwelack/msml_sofa
   ```

C.1.2. Running the container

There are two different options to run a graphical application inside the simulation container:

1. Start an SSH server inside the container and connect to it. This typically is a very stable set-up, but introduces an additional overhead.
2. Link the host X-server into the container. This approach can achieve bare metal performance, if the native graphics drivers are part of the container. However, it does not work on all host systems and has problems with Qt-based apps.

In the following we describe how to start the container in each mode. The start-up procedures described below are available in script form from Github:

```
$ git clone https://github.com/ssuwelack/msml-docker-runtime.git
```

[1] `www.docker.com`

Communicating to the container over SSH (recommended)

1. The following command starts the Docker container as a daemon, assigns it the name msml_sofa and binds it's SSH port to port 22000 of the localhost:

   ```
   $ docker run -d -p 127.0.0.1:22000:22 --name msml_sofa ssuwelack/
     msml_sofa /root/start_ssh.sh
   ```

2. Alternatively, the startup script can be used:

   ```
   $ ./start_ssh_msml_sofa.sh
   ```

3. We can use the pre-configured user msml (password: msml) to connect to the running container using SSH:

   ```
   $ ssh -XC msml@localhost -p 22000
   ```

4. The sofa runtime is located at /opt/sofa/bin. All data is stored in the home directory of the msml user.

5. The container can be stopped by executing

   ```
   $ docker stop msml_sofa
   ```

   on the host machine. It can be re-started using the command

   ```
   $ docker start msml_sofa
   ```

   Stopped containers can be deleted through

   ```
   $ docker rm msml_sofa
   ```

Communicating to the container via X-server sockets

In order to run the container with minimally overhead, the X-server socket can be forwarded. A start-up script is available in the above mentioned repository that carries out the necessary steps:

```
$ ./run_msml_sofa.sh
```

C.2. Connecting to a server instance

In order to avoid the work of setting up individual environments, the containers can be run on a server instance. In order to connect to such a container, you need the following information:

- Server name
- Public port number of the container
- Jump host name (if server is protected by a firewall)
- If not changed by the container provider, login credentials are:

  User: msml

  Password: msml

C.2.1. Connecting from a Linux client

In order to connect from a linux clients, just run:

```
$ ssh -XC msml@servername -p portnumber
```

C.2.2. Connecting from a Windows client

The MobaXterm framework can be used to connect from a windows machine. In this section, we describe how to run SOFA on a server instance. The server i61sv002.ira.uka.de exposes 25 docker container that run SOFA from port 22001 to 22025. In order to access these containers from a Linux machine inside the the HIS network, simply run

```
$ ssh -XC msml@i61sv002.ira.uka.de -p 22010
```

In the following we describe how the containers can be accessed from any Windows machine.

1. Install the MobaXterm framework which you can download on its homepage (http://mobaxterm.mobatek.net/download-home-edition.html).
2. Open MobaXterm and start a new session (cf. figure C.1).
3. Configure your SSH session (cf. figure C.2):
   - server name
   - user / password (standard is msml/msml)
   - port number
   - jump host for port forwarding and login credentials for jump host

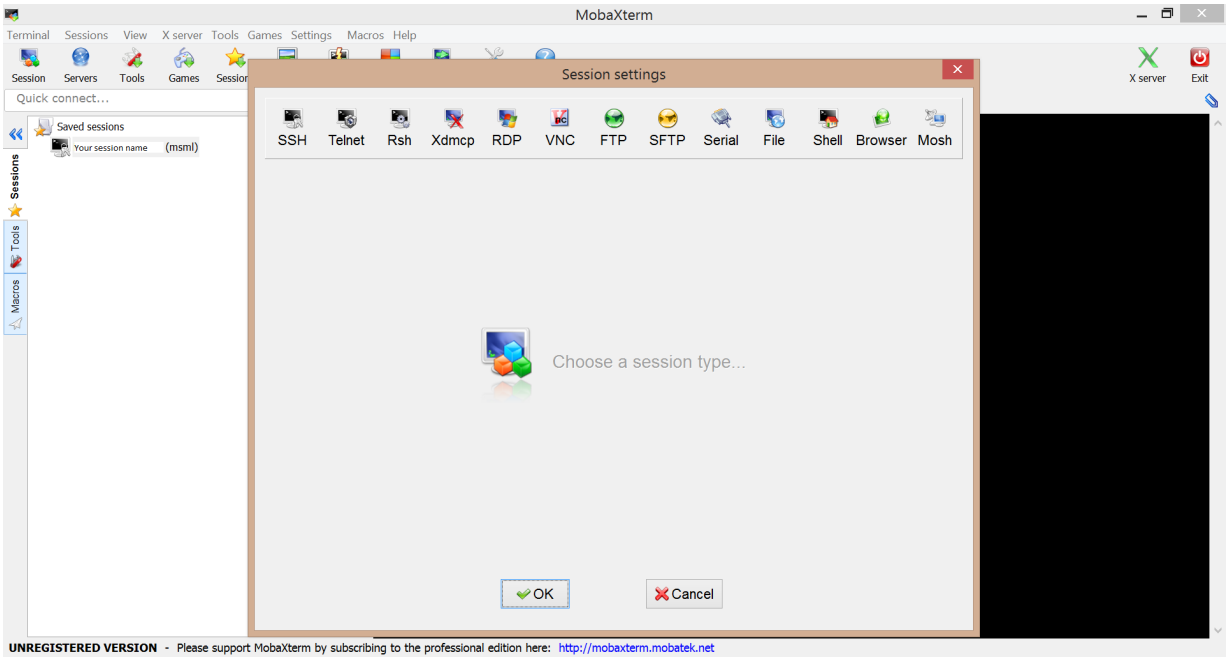


Figure C.1.: Screenshot of MobaXterm after starting a new session.

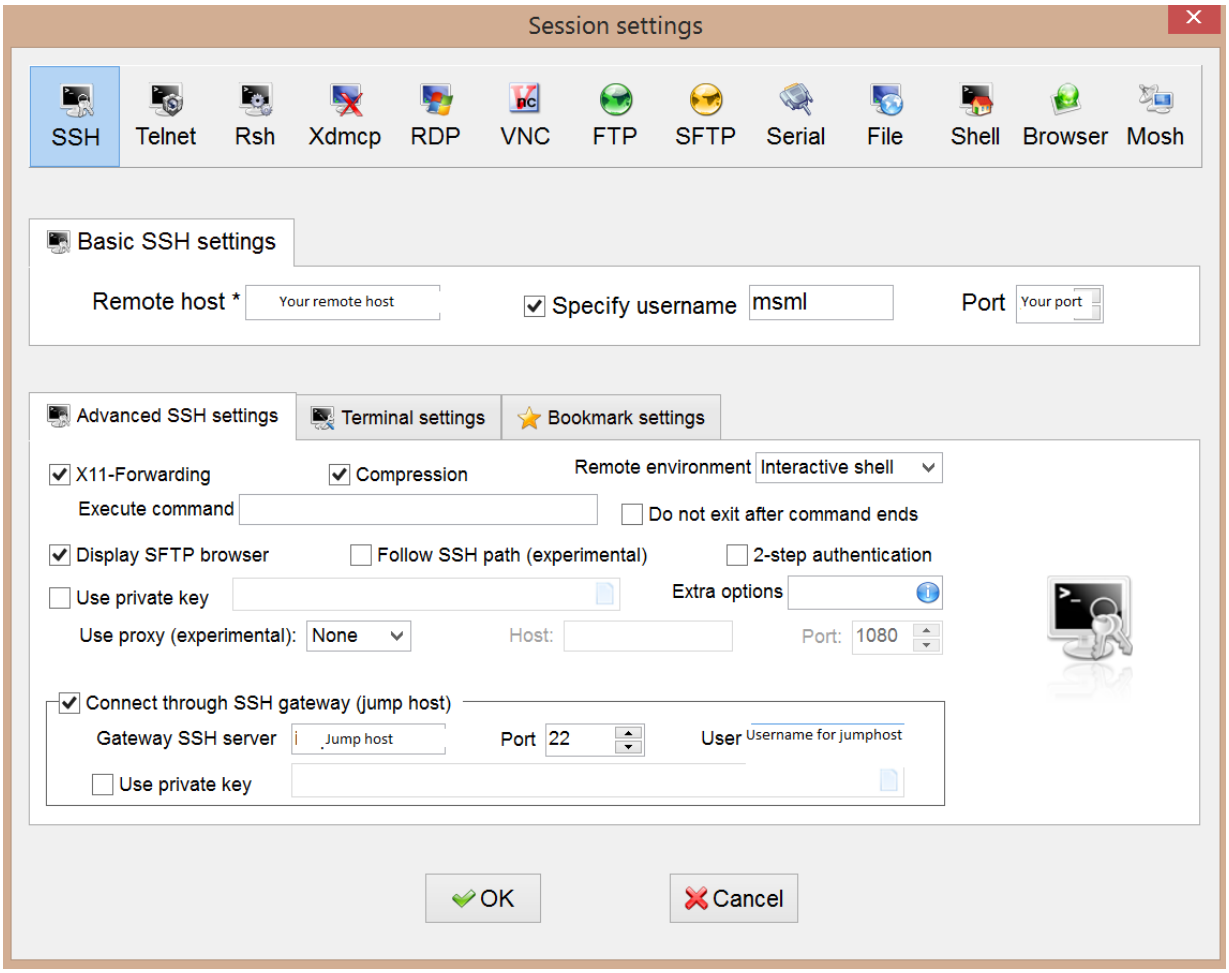


Figure C.2.: Screenshot of MobaXterm after starting a new session.

D.

Tutorial 1: Phenomenological Modeling

D.1. Pendulum with implicit Euler time integration

Start the SOFA scene file *Spring_EulerImplicit.scn* that describes a simple mass-spring system which is discretized using an implicit Euler scheme.
If you are using the pre-built Docker container this can be achieved by executing

```
$ cd /opt/sofa/bin
```

```
$ ./runSofa /home/msml/tutorials/Tutorial1/Spring_EulerImplicit.scn
```

In order to see the spring-mass model as shown Fig. D.1, please click *All* in the tab *View*. To animate a scene click on *animate*. For moving a particle press *shift* and click on the specified particle.

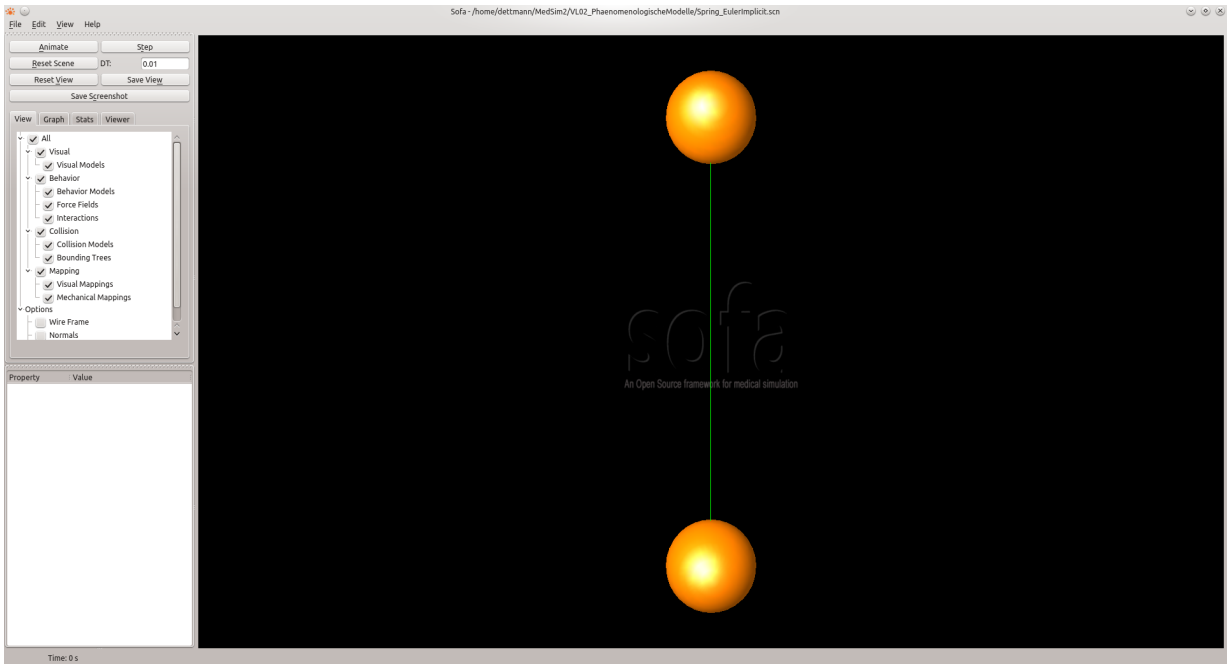


Figure D.1.: SOFA framework running a simple pendulum simulation.

Parametric studies can be used to get a better understanding of numerical methods. In order to analyze the implicit Euler technique, change the following input parameters

- time step size dt in s [0.001, 0.01, 0.1, 1]
- stiffness K_s and damping coefficient K_d of the spring (spring='Id1, Id2, K_s, K_d, L')

The parameters can be changed by opening the SOFA scene file (.scn) in a suitable text editor, e.g.

```
$ geany /home/msml/tutorials/Tutorial1/Spring_EulerImplicit.scn &
```

While changing the parameters observe the behavior of the simulation. In particular, try to answer the following questions

- How does the parameter effect the stability of the simulation?
- Does the frequency or amplitude change?
- How does the energy of the system behave over time?

D.2. Pendulum with explicit Euler time integration

The pendulum can also be discretized with an explicit Euler time integration scheme (*Spring_EulerImplicit.scn*). Repeat your analysis on this set-up. In particular, compare the behavior of the explicit and the implicit scheme.

E.

Tutorial 2: Simple Beam Model with MSML

The medical simulation markup language (MSML) is a flexible XML-based description for the biomechanical modeling workflow and finite-element based biomechanical models.
The example files include several scenarios based on a simple beam model. One scenario describes a linear elastic beam under gravity load. If you are using the pre-built Docker container you can open the scenario with

```
$ geany /opt/msml/examples/BeamExample/beamLinearGravity.msml.xml
```

In order to simulate the scenario the MSML exec command can be executed from the command line:

```
$ cd /opt/msml/src

$ ./msml.py exec -o outputfolderPath /opt/msml/examples/BeamExample/
    beamLinearGravity.msml.xml
```

To get the help information of MSML type the following:

```
$ ./msml.py -h (or for methods: $ ./msml.py exec -h)
```

ParaView can be used to view the results and inspect the meshes:

```
$ cd /opt/paraview/bin

$ ./paraview
```

After starting ParaView as shown in Fig. E.1 you have to load the output file (before you can press *play* you have to click on *Apply* in the tab *Properties*). The output mesh for each timestep is usually labeled 'dispx.vtu'.
The scenario settings (e.g. material parameters, boundary conditions) can be changed by opening the XML-file in a suitable text editor, e.g.

```
$ geany /opt/msml/examples/BeamExample/beamLinearGravity.msml.xml &
```

E.1. Using the Python API

This section describes how to create a running script for MSML in Python.
First open an editor (e.g. geany) and create a new file *myScript.py*. To run MSML from a python environment we need some imports as described in the following:

```
import os
import sys
sys.path.insert(0, envconf.MSML_ROOT)) #to use msml imports
import msml.api.simulation_runner as api
```

Then you have to define the input file and the output directory:

```
msml_infile = os.path.abspath("/opt/msml/examples/LiverExample/liverLinear.msml.xml")
msml_outdir = os.path.abspath("/tmp/MSMLResultsBeam/")
```

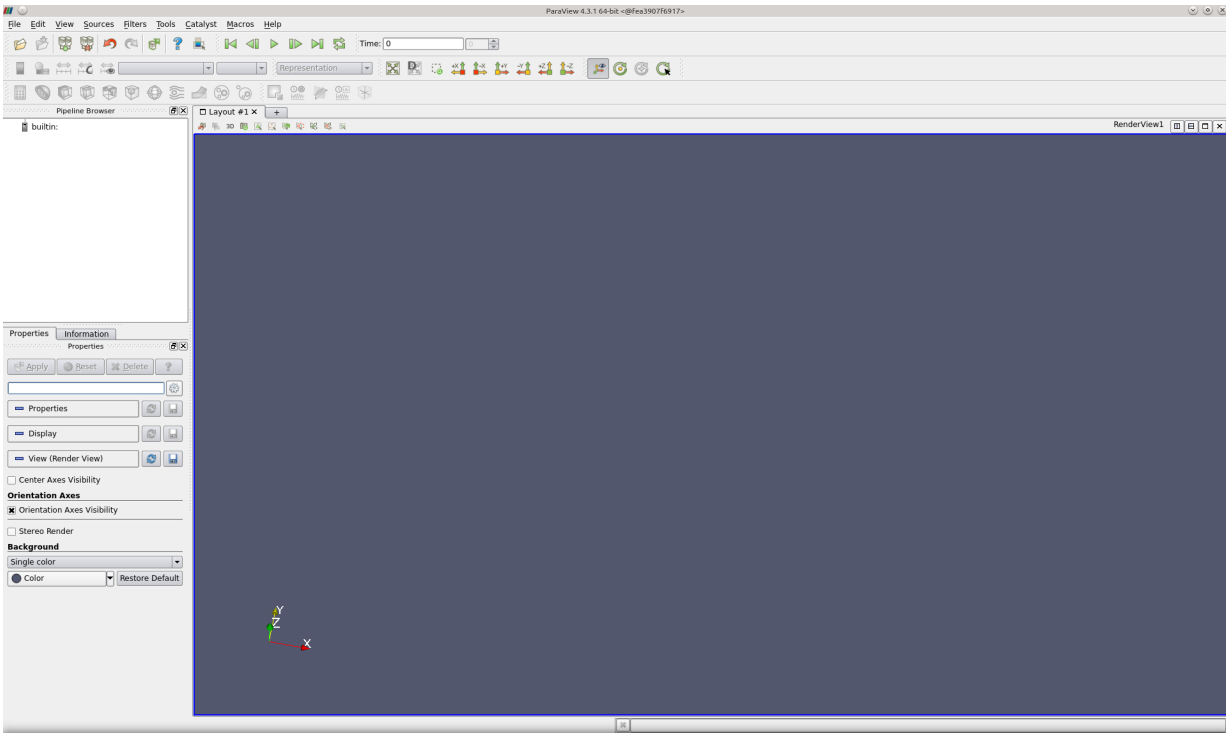


Figure E.1.: Start window of ParaView.

To start MSML the following code is necessary:

```
myRunner = api.SimulationRunner(msml_infile, "sofa", msml_outdir)
myRunner.run()
```

Now your script is complete. Start the script with

```
$ python myScript.py
```

F.

Tutorial 3: Convergence Analysis

Running a convergence analysis is a standard tool for validating numerical algorithms. The core idea is to first compute a reference solution using a known analytical solution or a high resolution gold standard numerical procedure. Subsequently, solutions with increasing resolutions are computed and compared to the reference solution. The error is usually visualized by means of a loglog plot in term of the number of degrees of freedom or the characteristic mesh size h.
The python API of MSML allows to quickly construct a convergence analysis. An python script example can be found in "/examples/ConvergenceAnalysis/Beam/ConvergenceAnalysis.py".
In order to run this example from the command line please first make sure that the MSML source is in your Python path:

```
$export PYTHONPATH="$PYTHONPATH:/opt/msml/src"
```

Then change to the folder that contains the script

```
$ cd /opt/msml/examples/ConvergenceAnalysis/Beam
```

and run it:

```
$python ConvergenceAnalysis.py
```

G.

Tutorial 4: Meshing

In order to run a soft tissue simulation a volume volume mesh has to be generated from tomographic data. MSML contains many different operators that can be combined into complex meshing pipelines that can be used for that purpose.
In this tutorial we show how to execute a simple pipeline that consists of a mesh reduction step followed by the actual volumetric meshing. We also highlight how the mesh quality of the resulting volume mesh can be inspected with the respective post-processing operators.
The python script example can be found in "/examples/MeshExample/Liver/liverMesh_runner.py". It relies on a pipeline that relies on the SCVD operator for mesh reduction. This operator only takes the number of the surface points in the reduced mesh as input. For volume mesh generation we use the TetGen mesher. In this tutorial we use the radius-edge ratio of the generated volume mesh as an input parameter. More information on this and more input parameters for the mesher can be found on the TetGen homepage (http://wias-berlin.de/software/tetgen/features.html).
As can be seen by opening the example script, we choose to generate a reduced surface mesh with 1800 points and generate a volume mesh with tetrahedra that have maximal radius-edge ratio of 1.4. Furthermore, the script calls MSML post-processing operators that compute various mesh quality measures. In this example the *edge ration*, the element *volume* as well as the *minimum angle* are considered and reported after mesh generation.
In order to run this example from the command line please first make sure that the MSML source is in your Python path:

```
$ export PYTHONPATH="$PYTHONPATH:/opt/msml/src"
```

Then change to the folder that contains the script

```
$ cd /opt/msml/examples/MeshExample/Liver
```

and run it:

```
$ python liverMesh_runner.py
```

ParaView can be used to view the result and inspect the mesh:

```
$ cd /opt/paraview/bin
```

```
$ ./paraview
```

By applying the filter function *Extract Cells By Region* as shown in Fig. G.1 the mesh can be visualized in Paraview.

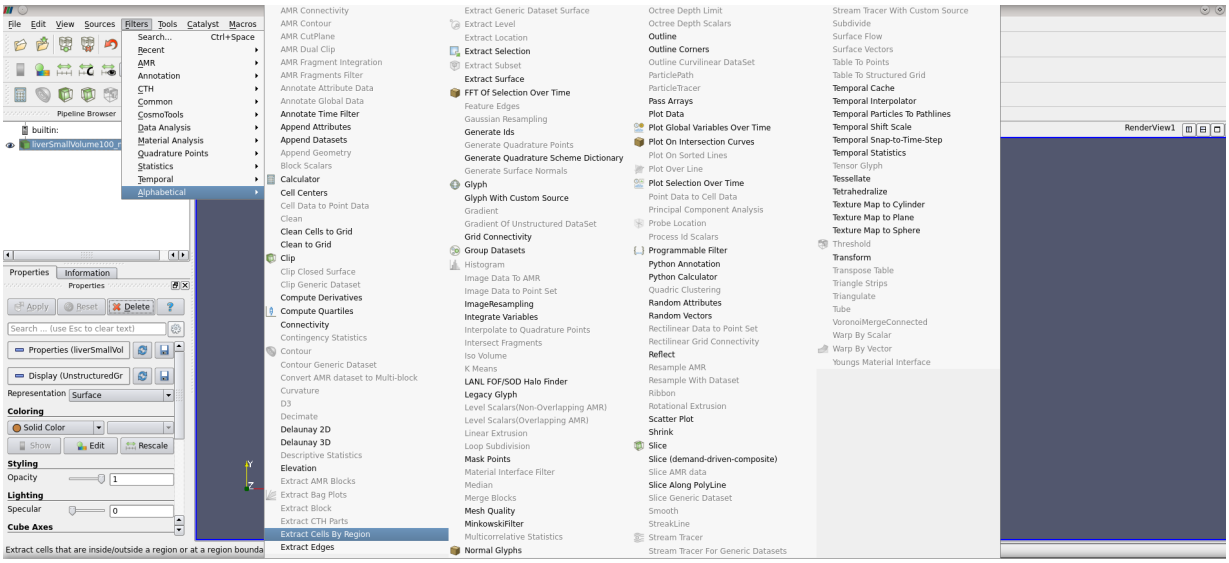


Figure G.1.: Get filter preferences in ParaView.

Bibliography

[BLM00] T. Belytschko, WK Liu, and B. Moran. *Nonlinear finite elements for continua and structures*, volume 36. Wiley, 2000.

[Bon97] Javier Bonet. *Nonlinear continuum mechanics for finite element analysis*. Cambridge university press, 1997.

[Bra07] Dietrich Braess. *Finite elements: Theory, fast solvers, and applications in solid mechanics*. Cambridge University Press, 2007.

[BW98] D. Baraff and A. Witkin. Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 43–54. ACM, 1998.

[CK92] AO Cifuentes and A. Kalbag. A performance study of tetrahedral and hexahedral elements in 3-d finite element structural analysis. *Finite Elements in Analysis and Design*, 12(3):313–318, 1992.

[FDD+12] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, et al. Sofa: A multi-model framework for interactive physical simulation. *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, pages 283–321, 2012.

[Fun93] Y.C. Fung. *Biomechanics: mechanical properties of living tissues*. Springer, 1993.

[H+94] Thomas JR Hughes et al. *Mathematical foundations of elasticity*. DoverPublications. com, 1994.

[HG01] G.A. Holzapfel and T.C. Gasser. A viscoelastic model for fiber-reinforced composites at finite strains: Continuum basis, computational aspects and applications. *Computer methods in applied mechanics and engineering*, 190(34):4379–4403, 2001.

[HN03] Stefan Hartmann and Patrizio Neff. Polyconvexity of generalized polynomial-type hyperelastic strain energy functions for near-incompressibility. *International Journal of Solids and Structures*, 40(11):2767–2791, 2003.

[Hol00] G.A. Holzapfel. *Nonlinear solid mechanics*. Wiley New York, 2000.

[Ibr09] Adnan Ibrahimbegović. *Nonlinear solid mechanics: theoretical formulations and finite element solution methods*, volume 160. Springer, 2009.

[Ogd97] Raymond W Ogden. *Non-linear elastic deformations*. Courier Dover Publications, 1997.

[RES10] S. Raghunathan, D. Evans, and J.L. Sparks. Poroviscoelastic Modeling of Liver Biomechanical Response in Unconfined Compression. *Annals of biomedical engineering*, 38(5):1789–1800, 2010.

[SLH+13] S. Suwelack, D. Lukarski, V. Heuveline, R. Dillmann, and S. Speidel. Accurate surface embedding for higher order finite elements. In *Symposium on Computer Animation*, 2013.

[SM03] Endre Süli and David F Mayers. *An introduction to numerical analysis*. Cambridge University Press, 2003.

[SRD+11] S. Suwelack, S. Roehl, R. Dillmann, A. Wekerle, H. Kenngott, B. Müller-Stich, C. Alt, and S. Speidel. Quadratic corotated finite elements for real-time soft tissue registration. In *MICCAI workshop: Computational Biomechanics for Medicine*. Springer, 2011.

[TCC+08] ZA Taylor, O. Comas, M. Cheng, J. Passenger, DJ Hawkes, D. Atkinson, and S. Ourselin. On modelling of anisotropic viscoelasticity for soft tissue simulation: Numerical solution and GPU execution. *Medical Image Analysis*, 2008.

[Wik13] Wikipedia. Wikipedia, the free encyclopedia, 2013. [Online; accessed 22-October-2013].

[ZT77] Olgierd Cecil Zienkiewicz and Robert Leroy Taylor. *The finite element method*, volume 3. McGraw-hill London, 1977.