

MASTER

FEM modeling and animation of human faces

Yu, C.

Award date:
2013

[Link to publication](#)

Disclaimer

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

FEM modeling and animation of human faces

Cong Yu

Department of Mathematics and Computer Science

Master thesis

Graduation committee

dr.Dmitry Znamenskiy (Philips)

prof.dr.ir. Barry Koren (TU/e, chairman)

dr.ir. Martijn Anthonissen (TU/e)

dr. habil. A. (Adrian) Muntean (TU/e)

Keywords: Facial soft tissue deformation; physically-based soft tissue model; hyperelastic material model; non-linear finite element method.

Abstract: Philips produces respiratory masks to treat apnoea (a sleep disorder) patients. In order to facilitate the development of the new masks, Philips Research creates computational tools for virtual mask fitting on 3D facial scans of the patients. Certain mask types are designed for patients who open their mouth during sleep, and therefore the virtual mask fitting of such types of masks requires 3D scans with the mouth open to a certain degree. Such 3D scans are difficult to collect. Within this context, the current work is focused on simulating the open mouth geometry from a patient's neutral closed mouth scan by means of a physically-based soft tissue model. The open-mouth (deformed) state is computed using finite element simulation software (Marc Mentat). The proposed model consists of a double-layer structure: skin layer and skull layer, with springs connecting them in between to simulate the sliding effect of superficial fascia. We assume the skull layer to be rigid and we presume different hyperelastic material models (St.Venant-Kirchhoff, Mooney-Rivlin and Gent) for the skin layer. Besides, a morphing method is also developed for benchmarking of the mouth-opening simulations. Finally, we compare all the methods and transform the most effective one into a mouth-opening tool.

Conclusions: We have developed five methods in total for the computational tool to open participants' mouths virtually. Except from the morphing method, the other four methods are based on 3D finite element soft tissue models (physically-based models) which allow a reasonable prediction of tissue deformations resulting from the open mouth process. The models are built based on individual facial surface scans and generic soft tissue thickness data. Using this information, we created our models in Lagrangian formulation. The models incorporate geometric nonlinearity and mechanical nonlinearity of the biological tissues. We verified our simulation results with the actual open mouth scans and calculated the average error and standard deviation map for each method. The methods based on the physically-based models generally outperform the morphing methods. For the best physically-based models, the largest error typically occurs at the cheek area and is within $[-2mm, 2mm]$.

Contents

1. Introduction	5
1.1. Objective.....	5
1.2. Problem description.....	6
1.3. Solution approach	6
1.4. Outline	7
2. State of the art of facial soft tissue modelling.....	8
2.1. Mass-spring models	8
2.2. FEM-based models	9
2.3. Other models	10
3. Background knowledge	11
3.1. Facial soft tissue structure and property	11
3.1.1. Facial soft tissue anatomy.....	11
3.1.2. General mechanical properties of soft tissue	13
3.1.3. Human facial soft tissue and its mechanical properties	14
3.2. Elasticity theory	16
3.2.1. Basic concepts in solid mechanics.....	17
3.2.2. Material constitutive equations (linear/hyper-elasticity).....	22
4. Mathematical models and implementation details.....	30
4.1. FEM-based facial soft tissue models	30
4.1.1. General model assumptions.....	30
4.1.2. Configuration of the model	31
4.1.3. Governing equations	32
4.1.4. Finite element approach	44
4.2. Morphing method (a computer vision-based method).....	63
4.3. Summary of methods	66
4.4. Implementation details	68
5. Results and discussion	77
5.1. Benchmarking I: RMS error.....	77
5.2. Benchmarking II: Average error map	79
5.3. Observations and discussions.....	84
5.3.1. Observations	84
5.3.2. Discussions.....	85
6. Conclusion and future work	87

7. Acknowledgement	90
A. Appendices.....	91
A.1 Model definition file.....	91
A.2 Linux subroutine	93
A.3 Marc procedure file.....	93
References	98

1. Introduction

Sleep apnoea is a sleep disorder when people frequently stop breathing during the night due to closure of the upper airway, so people partially wake up many times during the night, which causes continual sleepiness during the day and other health disorders. Sleep studies show that 6-7% of the Western population suffers from at least a mild form of apnoea, where almost 85% of the cases remain undiagnosed and untreated. Male gender, age, overweight, low muscle tone and snoring can increase the likelihood of apnoea up to 40%. Sleep apnoea can be effectively treated (but not cured) by providing positive air pressure which prevents the upper airway from obstruction. The air pressure is generated by a pump and delivered by means of a tube and a facial mask to a patient.

A good mask should satisfy two criteria: it should be comfortable, and should be air-tight. If the mask does not fit the face, a patient can get red marks where the mask contact is too tense, or air leaks where the mask contact is too loose. The air leaks reduce the efficiency of the therapy and, if the mask is leaking towards an eye, it can cause an eye inflammation. Traditionally, masks are evolved by a trial-and-error approach where numerous variations of the mask shapes are realized in silicone and tested on the live persons. In order to accelerate the optimization of the mask shapes Philips has a '3D mask sizing' project. In this project, Philips is developing a novel CPAP (Continuous Positive Airway Pressure) mask advice system which facilitates the professional mask advices given to the patients in sleep labs or DME (Durable Medical Equipment facility). Within the mask advice system, the patients are first scanned with a 3D scanner and then the system virtually fits the available masks to the 3D facial scans. Compared with the physical mask fitting, virtual mask fitting is less time consuming and expensive. Indeed, for physical mask fitting, if a mask does not fit, it cannot be fitted to another patient due to hygienic restrictions while the average price of a CPAP mask is about \$100. Since there are masks from different vendors and sizes (like small, medium and large) available, it is difficult for the doctor to pick the right mask for the patient. Thus, a virtual mask fitting system provides a solution for the doctor and the patient who can virtually try several masks at no cost. Based on all the advantages listed above, the virtual mask advice system can increase the percentage of the patients who are satisfied with the therapy and continue the treatment. The patients complying with the therapy generate a significant part of income of Philips Respironics since the mask has the life time of 3-6 months after which a new mask should be purchased.

1.1. Objective

The virtual mask fitting system requires accurate and realistic models for quantitative assessment of the patient comfort for different mask types. For a patient who opens his/her mouth during sleep, a full face mask which covers both the mouth and the nose is needed to provide proper pressure through the mouth and nose. see Figure 1.1. The advice of the full face masks should be naturally based on the open mouth 3D facial scans. However, people do not correctly interpret the 'open mouth' (to the extent during their sleep) command. In Figure 1.2, we show the four open mouth sample scans. One can see that the participants open their mouths in different ways. The left-most one opened her mouth really widely while the right most man's mouth opening is barely visible.

Motivated by above facts, the main objective of my graduation project is to "open" the participants' mouths in a uniform way given his/her neutral (closed-mouth) facial scan and possibly other information about the face (e.g. the thickness of the soft tissue). In other words, I am going to develop a computational tool which gives reasonably accurate prediction on the open mouth geometry based on the closed mouth scans and the desirable open mouth angle.



Figure 1.1 An application of a typical CPAP full face mask



Figure 1.2 Sample open mouth scans

1.2. Problem description

To simulate the mouth opening process, we need a reliable human face model with suitable mathematical boundary conditions. To derive a satisfactory model, we have to address the following two sub-problems:

- How to do the geometric modelling
- How to do the mechanical modelling

The geometric modelling involves the capturing of complex facial surface feature, mesh generation and the processing of volumetric data (for example the soft tissue thickness information obtained from CT or MRI data) from the surface processing.

The mechanical model consists of a proper constitutive relation (which means that it can reasonably well represent the material properties), a suitable physical conservation law to obey, and appropriate boundary conditions.

1.3. Solution approach

The overview of our simulation pipeline is depicted below. Detailed explanations on each step will be given in Section 4.4.

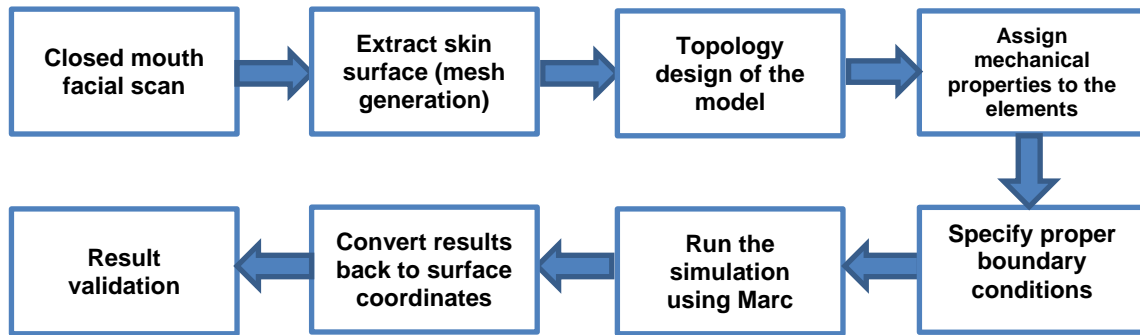


Figure 1.3 Solution approach

1.4. Outline

The thesis is organized as follows: Section 2 gives a review of the state-of-art in soft tissue modelling. Then some useful biological and continuum mechanical background knowledge is presented in Section 3. We will give a detailed mathematical description on the models developed in this project and implementation details in Section 4. Next, simulation results and discussions are presented in Section 5. Conclusion and suggestions for further work will be given in the last section.

2. State of the art of facial soft tissue modelling

A considerable amount of studies on facial soft-tissue modelling have been performed over the past three decades. Early works such as [1] restricted themselves to pure geometric deformations (so-called non-physical models). However, with physically based modelling, more realistic facial models arose. First, it was based on mass spring systems and finite difference schemes. Later researchers applied the finite element method for the solution of the related partial differential equations. Below we will only focus on the important physically based models.

Depending on the computational methods, most models fall into three main categories: mass-spring models (MSM), models based on a finite element method (FEM), and other models.

2.1. Mass-spring models

Mass-spring models consist of a set of mass points linked by springs and possibly dampers. In the simplest formulation, the equation of motion of a point i is:

$$\mu \frac{d^2 \vec{r}_i}{dt^2} = -\gamma \frac{d\vec{r}_i}{dt} + \sum_{j \in N(i)} K_{i,j} \frac{(l_{i,j}^0 - \|\vec{r}_{ij}\|) \vec{r}_{ij}}{\|\vec{r}_{ij}\|},$$

where \vec{r}_i is the displacement vector of node i . $\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$ is the direction vector. μ is the mass, γ the damping factor and $K_{i,j}$ the stiffness of the spring connecting point i and points j in the neighbourhood $N(i)$ of point i .

MSM were widely accepted for real-time application over the last twenty-five years. Early work focused on facial animation based on the masses and the springs. Waters [2] defined springs on layered regular lattices and used biphasic springs with two stiffness parameters to match the stress/strain curve of soft tissue described in a biomechanical study. Keeve et al. [3] deployed a similar approach for modelling fat tissues, but with the addition of a volume-preservation force intended to model the incompressibility of human soft tissue. Later, Keeve et al. [4] proposed another model with the assigned spring stiffness which depends on the elasticity of the soft tissue type, layer thickness and the size of the tissue elements. In [5], an alternative formulation of mass-spring systems was presented, where the model enables the user to specify the anisotropy independently of the geometry of the underlying mesh. Moreover, a method for generating constant volume deformation was proposed.

The main advantages of spring models over finite element based models are their ease of implementation (since they do not require continuous parameterization) and their computational efficiency. However, they suffer the following problems in soft tissue modelling [6]:

- **Topological design**
The topology of the mass-spring models is very important. Because under-constrained systems will fall into unwanted local minima, while over-constrained systems will decrease the range of deformation.
- **Validity of deformations**
The deformation results induced by springs-mass system cannot be easily compared to those given by biomechanical studies because springs do not rely on continuum mechanics. Compared with FEM models, one of the major disadvantages of mass-spring models is the inability to increase the precision of the model using the same mesh structure (e.g. by doubling the amount of nodal points in the model).

Below shows a typical mass-spring model structure:

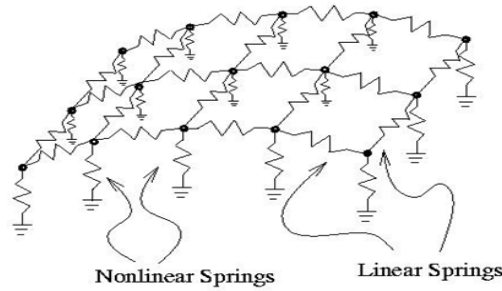


Figure 2.1 Mass-spring model structure

2.2. FEM-based models

The finite element method is widely used in the engineering world. FEM-based models are superior to most other models when accurate solution of continuum mechanics problems with complex geometry has to be found. It also provides the most flexible modeling platform free of all limitations with respect to the material type and the boundary conditions. FEM has strong biomechanical relevance at the cost of high computational effort, especially for complex geometry and large deformations.

Unlike mass-spring models starting with a discrete object model, FEM-based models consider a deformable object as a continuum: a solid body with mass and energies distributed throughout. FEM is used to find an approximation for a continuous function that satisfies some equilibrium equations. In FEM, a body is discretized as a set of basic elements (triangles, quadrilaterals, tetrahedra, prisms, etc.) on which shape functions with compact support are defined. This leads to continuous representation with varying levels of continuity. A finite element model is represented by the node displacement vector u . The equilibrium equations together with stress-strain relationship lead to the equation $f(u) = 0$. This equation can also be derived by the principle of minimization of the strain energy or by the principle of virtual work. Below, a review of existing and relevant FEM-based soft tissue models is given.

Back in 1986, Larrabee [7] presented a finite element model of skin deformation. The work is followed by Deng's PhD thesis [8]. Later, Kerve et al. [9] proposed an anatomy-based linear finite prismatic element model which is based on CT volume data and laser scans. Koch et al. [10] simulated facial surgery using a surface-based FEM model with springs (combined with struts spring structure refinement) to represent the soft tissue layer. However, this model lacks the true volumetric physics although the stiffness of the springs is derived from CT-scan images. By employing a similar approach but adding a muscle model, an emotion editor is created in [11]. A true volumetric FEM model was proposed in [12] for facial surgery simulation with the capability of incorporating incompressibility. However this model was restricted to the linear elastic theory and thus not suitable for large deformations. To achieve higher accuracy, Roth et al. [13] developed the tetrahedral Bernstein-Bézier elements by combining the finite element method with Bernstein-Bézier representation. Later they upgraded the globally C^0 -continuous surface to the C^1 -continuous model [14]. Nebel [15] introduced a soft tissue model which only relies on 3D scan data by presuming generic soft tissue structure shape and mechanical parameters. In [16], Cotin presented three different models, based on linear elasticity theory and the finite element method, which are well suited for surgery simulation with a strong focus on real-time application.

Since linear elasticity approximates the deformation accurately only when the deformation is small, researchers started to develop FEM-based models using non-linear elasticity theory. Guillaume et al. [17] implemented St.Venant-Kirchhoff elasticity and incompressibility constraints. A comparison was carried out between results (displacement in the chin area resulting from bone realignment) computed with a linear elastic model, a geometrical nonlinear model and a physical nonlinear model in E. Gladilin's work [18].

To model the anisotropic effect, Chabanas et al. [19] presented a layered element structure model, in which they assigned different mechanical properties to elements on different layers (dermis and hypodermis). Wang and Yang [20] first investigated the hyperelastic models of the facial soft tissues and then proposed a novel non-linear biomechanical model based on mixed elements, which is designed to address the heterogeneity in geometry and mechanical properties. Hung et al. [21] presented a four-layer model, with all layers treated as non-linear isotropic Mooney Rivlin materials, which aims at simulating wrinkling in facial soft tissue composite. In Bledie et al. [22], a finite element facial model was created, based on patient specific bone and skin image data, combined with generic muscle data employing commercial FEM software.

Simulation of deformable objects becomes a hot topic in biomedical engineering, computational physics, and computer graphics. Except from mass-spring models and FEM-based models, there are a lot of other models that have been proposed over the past decades:

2.3. Other models

1) *Mass-tensor model (MTM)*

MTM was originally developed to find a golden mean between speed and accuracy, later it was extended to non-linear, anisotropic elasticity. Recently, linear MTM was successfully applied to cranio-maxillofacial surgeries [16].

2) *Boundary Element Method-based models (BEM)*

BEM is a physically accurate method restricting the computation domain on boundaries. It computes the deformation by numerically solving the boundary integration equation on a surface mesh. Since the global matrix of BEM is dense and asymmetric and needs to be updated each step, James and Pai [23] use a multi-resolution Green's Function to accelerate the computation of the BEM matrix. In [24], BEM is coupled with a mass-spring constraint model. The work is devoted to provide visually accurate results for surgical trainees and to work efficiently in a real-time simulator.

3) *Molecular model [25]*

This method can be seen as a generalized mass-spring model, where mass points are, in fact, spherical mass regions called molecules. Elastic forces are then established between molecules by a spring-like connection. [23] also aims at integrating properties of materials to define the stiffness of such spring-like connections.

3. Background knowledge

To build a realistic facial tissue model, knowledge from the following fields of study is needed.

- Biological knowledge
- Elasticity theory

This section consists of two subsections: The first subsection gives a basic introduction to the anatomy of human facial soft tissue and its mechanical properties. The second subsection covers the basics in elasticity theory which will be used in the mathematical description of the model in Section 4.

3.1. Facial soft tissue structure and property

3.1.1. Facial soft tissue anatomy

Facial soft tissue includes various structures: skin, muscle, fascia vasa, nerves etc.. Below we will give a brief description on skin, muscle and fascia.

Skin

The integument or skin is the largest organ of the body, making up 16% of body weight, with a surface area of 1.8 m^2 . It has a multicomponent microstructure, the basis of which is an intertwined network of collagen, nerve fibers, small blood vessels and lymphatic, covered by a layer of epithelium and transfixated at intervals by hairs and the ducts of sweat glands, as illustrated in Figure 3.1. Skin is a dynamic organ in a constant state of change, as cells of the outer layers are continuously shed and replaced by inner cells moving up to the surface. Although structurally consistent throughout the body, skin varies in thickness according to anatomical site and age of the individual. Skin varies in thickness over the face: around the eyes it is thin, whereas around the lips it is thick. Skin thickness is less in women than in men, giving a luster to the skin surface of women.

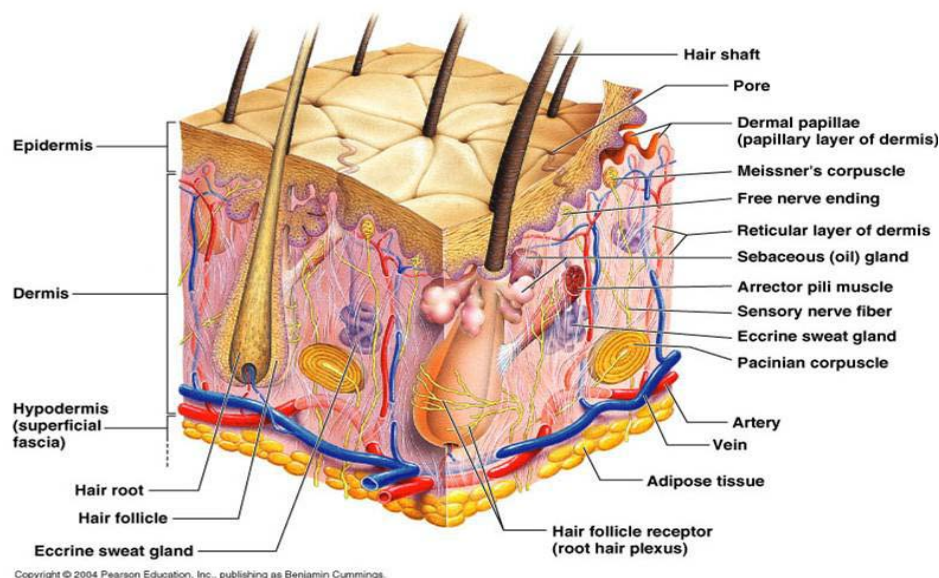


Figure 3.1 Typical human skin structures

Human skin has a layered structure consisting of the epidermis and the dermis. The epidermis is the outer layer, serving as the physical and chemical barrier between the interior body and exterior environment; the dermis is the deeper layer providing the structural support of the skin.

- The epidermis

The epidermis is stratified squamous epithelium. The main cells of the epidermis are the keratinocytes, which synthesize the protein keratin. The four separate layers of the epidermis are formed by the differing stages of keratin maturation. The epidermis varies in thickness from 0.05 mm on the eye lids to 0.8±1.5 mm on the soles of the feet and palms of the hand. Moving from the lower layers upwards to the surface, the four layers of the epidermis are:

1. stratum basale (basal or germinativum cell layer)
2. stratum spinosum (spinous or prickle cell layer)
3. stratum granulosum (granular cell layer)
4. stratum corneum (horny layer)

- The dermis

The dermis consists of irregular, moderately dense, soft connective tissue. Its matrix consists of an interwoven collagenous network, with varying content of elastin fibers, proteoglycans, fibronectin and other matrix components, blood vessels, lymphatic vessels and nerves. The dermis varies in thickness, ranging from 0.6 mm on the eye lids to 3 mm on the back, palms and soles. It is found below the epidermis and is composed of a tough, supportive cell matrix. Two layers comprise the dermis: a thin papillary layer and a thicker reticular layer.

The papillary dermis lies below and connects with the epidermis. It contains thin loosely arranged collagen fibers. Thicker bundles of collagen run parallel to the skin surface in the deeper reticular layer, which extends from the base of the papillary layer to the subcutaneous tissue. The dermis is made up of fibroblasts, which produce collagen, elastin and structural proteoglycans, together with immune-competent mast cells and macrophages. Collagen fibers make up 70% of the dermis, giving it strength and toughness. Elastin maintains normal elasticity and flexibility while proteoglycans provide viscosity and hydration. Embedded within the fibrous tissue of the dermis are the dermal vasculature, lymphatic, nervous cells and fibers, sweat glands, hair roots and small quantities of striated muscle.

Fascia

There are two types of fascia existing in facial soft tissue: superficial fascia and deep fascia. Underneath skin is a loose connective tissue layer. This is the superficial fascia (also called the subcutaneous or hypodermis). It consists of adipose tissue distributed in a network of connective fibers. This connective tissue is mostly collagen arranged in a lattice with fat cells. Thus it is an important depot of fat. For different age, gender or different places on face, it contains different levels of fat. For example, between males and females there is a difference in the underlying fat tissue, there being less in men than in women. It usually has the least fat at forehead and nose and the most at cheeks. Superficial fascia stores fat, anchors the skin to the underlying structures (mostly to muscles), but loosely enough that the skin can slide relatively freely over those structures.

Beneath the superficial fascia lies the deep fascia, which coats the bones or other tissues. It is also the landing point of certain muscles.

Muscle

The muscles of the face are commonly known as the muscles of facial expression. Some facial muscles also perform other important functions, such as moving the cheeks and lips during

chewing, swallowing, and speech. The muscles of facial expression are superficial, and all attach to a layer of subcutaneous and skin at their insertion. Some of the muscles attach to skin at both the origin and the insertion. Generally, three types of muscle can be discerned as the primary motion muscles: linear/parallel muscles, elliptical/circular sphincter-type muscles and sheet muscles.

The main movement involved in mouth opening is the mandible motion. In this complex process, muscles work in groups with other muscles to perform a smooth, balanced, coordinated series of movement of the mandible. Figure 3.2 illustrates the muscle distribution over the lower face.

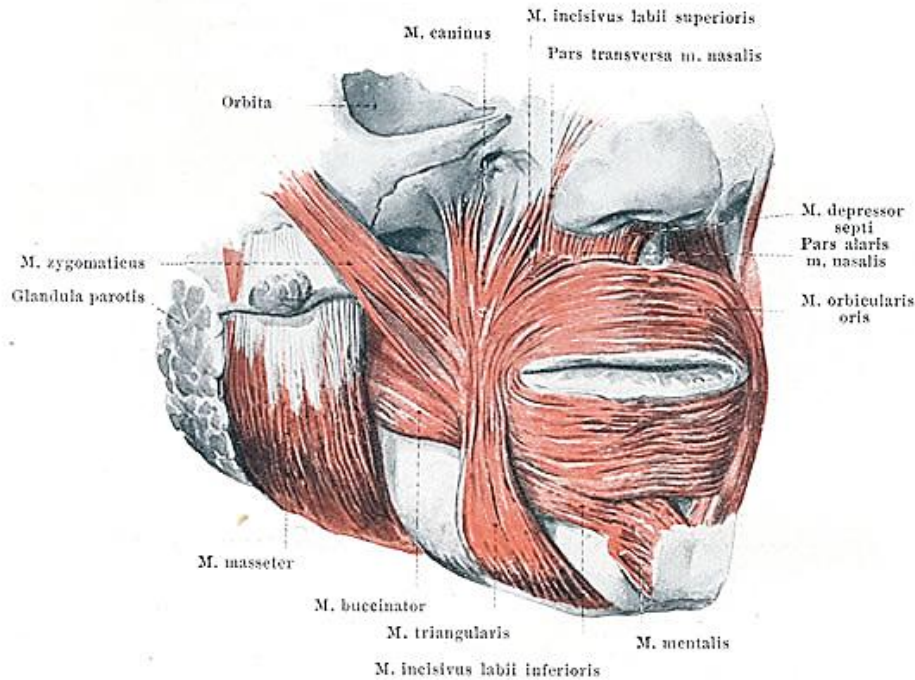


Figure 3.2 Lower face muscle distribution

3.1.2. General mechanical properties of soft tissue

There are numerous publications about biomechanical properties of soft tissues. Summarizing the facts observed from various experiments with different tissue types, soft tissue generally exhibits nonhomogeneous, anisotropic, quasi-incompressible, non-linear and plastic-viscoelastic material properties [28]. Below I am going to explain those properties one by one.

- **Heterogeneity and anisotropy**
Soft tissues are multi-composite materials containing cells, intracellular matrix, fibrous and other microscopic structures. Consequently, the mechanical properties of living tissues, e.g. material stiffness and compressibility, depend on the spatial direction. But since the heterogeneity and the anisotropy of microscopic structures can hardly be obtained from the available data, these properties are seldomly considered in a discrete macroscopic model.
- **Plasticity**
In physics, plasticity describes the large deformations of a material which undergoes irreversible energy dissipation and material destruction. Largely deformed soft tissue shows plastic behaviour. It is worth pointing out that during the mouth-opening process there is no deformation large enough to cause plastic behaviour, therefore the plasticity property is not relevant in the present project.
- **Viscoelasticity**

The time-dependent material behaviour is called viscoelasticity. Different from a purely elastic material, a viscoelastic solid does not respond to loading and unloading immediately, but with an exponential delay, which is known as creep and recovery. For the purpose of our project, we do not take into account this property, since we are only interested in its static state.

- Quasi-incompressibility

A material is called incompressible if its volume remains unchanged by the deformation. Soft tissue is a composite material that consists of both incompressible and compressible ingredients. Due to the high proportion of water, soft tissue is assumed to be quasi-incompressible (not totally incompressible but close). In this project, we describe facial tissue as a quasi-incompressible material and further treatments on quasi-incompressibility will be given in Section 4.

- Nonlinearity

There are two kinds of nonlinearities, which are relevant for the modelling of deformable soft tissue. The nonlinear relationship between strain and displacement is known as the geometrical nonlinearity. This type of nonlinearity is an intrinsic property of any mechanical continuum independently from its material properties. Another type of nonlinearity is caused by the nonlinear stress – strain relationship. This kind of nonlinearity is called the physical nonlinearity.

Figure 3.3 illustrates intuitively the nonhomogeneity of human skin by showing the complex anatomic structure in the nose and paranasal sinuses cross section:

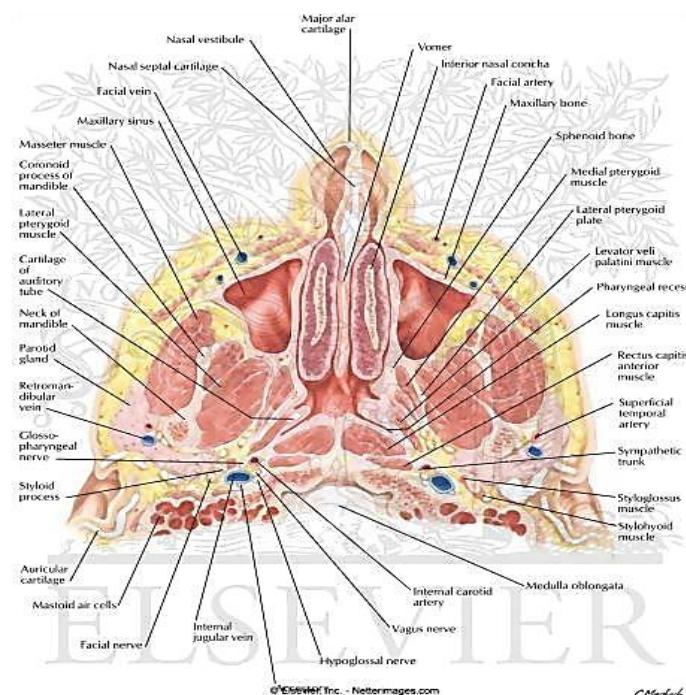


Figure 3.3 Nose and paranasal sinuses: cross section [26]

3.1.3. Human facial soft tissue and its mechanical properties

After discussing the general properties of soft tissue, we limit our focus on the facial soft tissue. Just as pointed out in the human facial anatomy section, the facial composite is logically separated into layers (epidermis, dermis, superficial fascia and muscle) due to the significant difference in mechanical properties between the layers.

The epidermis is thin (around 2.5% of the dermis thickness) and comprised of dead cells. It is considered to be insignificant in terms of mechanical properties. The dermis is primarily responsible for the mechanical properties of skin as a result of the interaction of two cellular bases: collagen (about 75% of skin dry weight) and elastin (about 4% of skin dry weight).

Collagen, the basic structural element for soft and hard tissues in animals, gives mechanical integrity and strength to our bodies. It has a strong stress response to applied load and has a limited range of deformation. Collagen fibers are the main constituent of the dermis and form an irregular network of wavy coiled fibers that run almost parallel with the skin surface [29]. Collagen is characterized by high strength (tensile strength of 1.5-3.5MPa), low extensibility (rupture at strains in the order of 5-6%), and high stiffness (Young's modulus approximately 0.1GPa to 1GPa in the linear region). The width of the bundles is 1-40 μ m [30].

Elastin fibers are the second main component of the dermis. Elastin is the most “linearly” elastic bio-solid material known. It is less stiff than collagen. The fiber width is 0.5-8 μ m. The behaviour of elastin is very similar to an ideal rubber with an essentially linear stress/strain response over a wide range of deformations [29]. The stress-strain curve of elastin is shown in Figure 3.4:

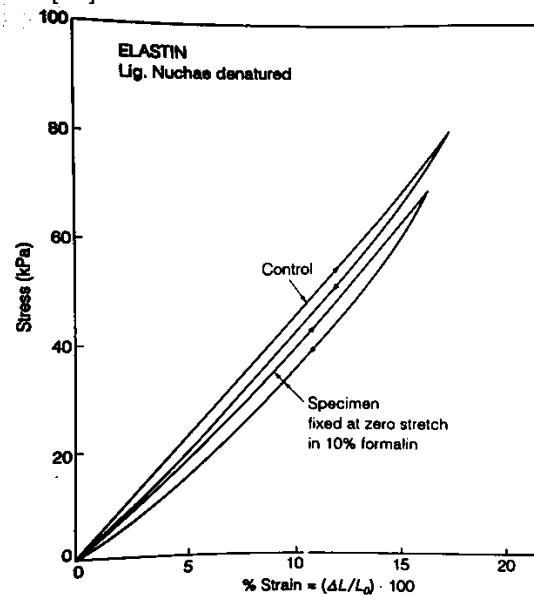


Figure 3.4 the stress-strain curve of elastin [29]

Figure 3.5 gives the strain/stress curve of human skin in uniaxial tension case:

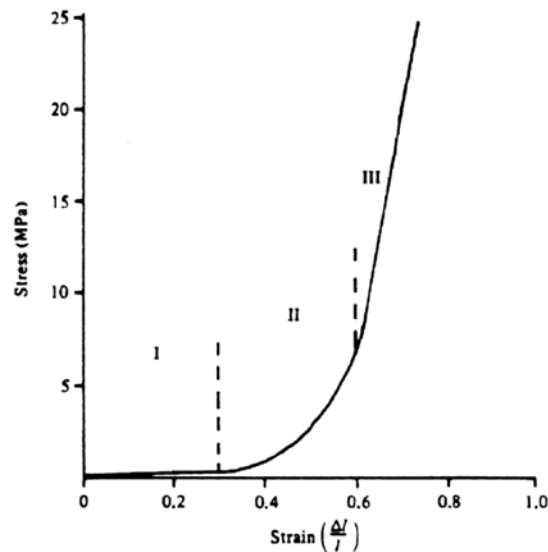


Figure 3.5 Stress-strain relationship of skin in different stages [31]

The stress-strain curve of skin for uniaxial tension is nonlinear due to the non-uniformity of its structure, as can be seen in Figure 3.5. The curve can be divided into four stages. In the first stage the contribution of response of the undulated collagen fibers can be neglected and elastin is mainly responsible for the skin stretching. Thus the stress-strain relation is approximately linear with a Young's modulus of approximately 5kPa according to Daly [32] (see Figure 3.6), since elastin can be considered as a "linear" elastic material. In the second phase, a gradual straightening of an increasing fraction of the collagen fibers causes an increase in stiffness. In the third phase all collagen fibers are straight and the stress-strain relation becomes linear again. Beyond the third phase, rupture of the fibers occurs.

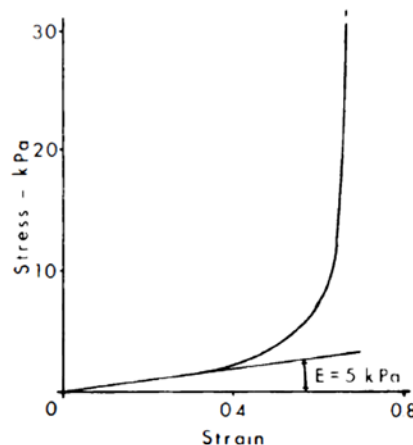


Figure 3.6 Stress-strain curve of human skin at low stress [32]

3.2. Elasticity theory

In the biological world, atoms and molecules are organized into cells, tissues, organs and individual organism. At the atomic and molecular level the movement of matter must be analysed with quantum, relativistic, and statistical mechanics [29]. At the cellular, tissue, organ and organism level, it is usually sufficient to take Newton's law of motion as an axiom. In this context, we consider the tissue level. In these systems it is convenient to consider the material as a continuum. Below, the several fundamental solid mechanics concepts are presented which will

be crucial in the later mathematical analysis of the FEM-based soft tissue models.

3.2.1. Basic concepts in solid mechanics

The basic solid mechanics concepts we are going to introduce include: Cauchy stress tensor, strain, Green's strain tensor, Cauchy's infinitesimal strain tensor, left and right Cauchy-Green deformation tensor, internal virtual work and the second Piola-Kirchhoff stress tensor.

Cauchy stress tensor

Let the cross-section of a body be A , and let the force that acts in the tissue be F . The ratio

$$\sigma = \frac{F}{A}$$

is the *stress* in the body. The basic unit of stress is Newton per square meter (N/m^2) or Pascal (Pa). $1 \text{ Pa} = 1 \text{ N/m}^2$. We define the *stress vector*

$$\mathbb{T}_v = \frac{dF}{dS},$$

where a subscript v is used to indicate the relevance of the normal direction of the surface dS . Thus vector \mathbb{T}_v represents the force per unit area acting on the surface.

Imagine a little cube in the body. Consider a set of rectangular Cartesian coordinates x_1, x_2, x_3 . Let the surface of the cube normal to x_1 be denoted by ΔS_1 . Let the stress vector that acts on the surface ΔS_1 be T_1 . Resolve T_1 into three components in the direction of the coordinate axes and denote these by $\sigma_{11}, \sigma_{12}, \sigma_{13}$. Similarly, we treat the x_2, x_3 directions. The 3×3 matrix σ_{ij} is defined as *Cauchy stress tensor*. Below is an illustration about the Cauchy stress tensor:

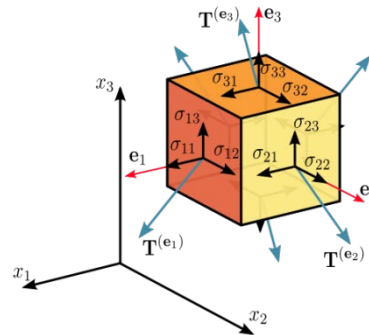


Figure 3.7 Cauchy stress tensor illustration

Cauchy observed that the stress vector \mathbb{T} across a surface will always be a linear function of the surface's normal vector v , the unit-length vector that is perpendicular to it. That is, $\mathbb{T} = \sigma(v)$, where the function σ satisfies

$$\sigma(\alpha v_1 + \beta v_2) = \alpha \sigma(v_1) + \beta \sigma(v_2),$$

for any vectors v_1, v_2 and any real numbers α, β . With the fundamental laws of conservation of linear momentum and static equilibrium of forces, we derive the linear relation between \mathbb{T} and v .

$$\mathbb{T} = \sigma * v,$$

where “ $*$ ” is matrix multiplication. Note that it can be shown that if a body satisfies rotational equilibrium, then its stress tensor is symmetric.

Strain

Deformation of a solid is described by strain. Take a string of initial length L_0 , and its stretched length L . Use dimensionless ratios to eliminate the absolute length from consideration, then the following *strain measures* can be defined:

$$\epsilon_1 = \frac{L - L_0}{L_0}, \quad \epsilon_2 = \frac{L - L_0}{L}.$$

Other strain measure examples include:
Green strain:

$$\epsilon_g = \frac{L^2 - L_0^2}{2L^2},$$

and Euler-Almansi strain:

$$\epsilon_e = \frac{L^2 - L_0^2}{2L_0^2}.$$

It is obvious that in infinitesimally small elongations all the above strain measures are asymptotically equal. However, in finite elongations, they are different.

Green strain tensor

To adopt a more general treatment of the matter, we introduce the concept of *strain tensor*. Below, we will give the derivation of Green strain tensor.

Firstly, a particle P in a body located originally at a place with coordinates (X_1, X_2, X_3) is moved to the place with coordinates (x_1, x_2, x_3) , when the body deforms. The vector u is called the displacement vector of the particle. Clearly,

$$u_1 = x_1 - X_1, \quad u_2 = x_2 - X_2, \quad u_3 = x_3 - X_3.$$

The deformation of the body is known if x_1, x_2, x_3 are known functions of X_1, X_2, X_3 and time t :

$$\vec{x} = \phi(\vec{X}, t),$$

where $\vec{x} = (x_1, x_2, x_3)$, and $\vec{X} = (X_1, X_2, X_3)$. We assume that the transformation is one-to-one. Namely, the function above has a unique inverse,

$$\vec{X} = \phi^{-1}(\vec{x}, t),$$

for every point in the body.

To study the stretching and distortion of the body, it is sufficient if we know the change of distance between any arbitrary pair of points. In the original configuration, consider an infinitesimal line element connecting the point $P(X_1, X_2, X_3)$ and $P'(X_1 + dX_1, X_2 + dX_2, X_3 + dX_3)$. The length ds_0 of PP' in the original configuration is given by:

$$ds_0^2 = dX_1^2 + dX_2^2 + dX_3^2.$$

When P and P' are deformed to points $Q(x_1, x_2, x_3)$ and $Q'(x_1 + dx_1, x_2 + dx_2, x_3 + dx_3)$, respectively, the distance of the length ds of the new element QQ' is

$$ds^2 = dx_1^2 + dx_2^2 + dx_3^2.$$

By the deformation function, we have

$$\begin{aligned} dx_i &= \frac{\partial x_i}{\partial X_j} dX_j, & dX_i &= \frac{\partial X_i}{\partial x_j} dx_j, \\ ds_0^2 &= \delta_{ij} \frac{\partial X_i}{\partial x_l} \frac{\partial X_j}{\partial x_m} dx_l dx_m, & ds^2 &= \delta_{ij} \frac{\partial x_i}{\partial a_l} \frac{\partial x_j}{\partial a_m} dX_m dX_l. \end{aligned}$$

Note that repetition of an index in a single term means a summation over the whole range of index and this notation will be used later throughout the report whenever a summation symbol is not used. Hence, the difference between the square of the length elements may be written as:

$$\begin{aligned} ds^2 - ds_0^2 &= \left(\delta_{\alpha\beta} \frac{\partial x_\alpha}{\partial X_i} \frac{\partial x_\beta}{\partial X_j} - \delta_{ij} \right) dX_i dX_j, \\ ds_0^2 - ds^2 &= \left(\delta_{ij} - \delta_{\alpha\beta} \frac{\partial X_\alpha}{\partial x_i} \frac{\partial X_\beta}{\partial x_j} \right) dx_i dx_j. \end{aligned}$$

We define the *Green strain tensor* E and *Almansi strain tensor* e as:

$$\begin{aligned} E_{ij} &= \frac{1}{2} \left(\frac{\partial x_\alpha}{\partial X_i} \frac{\partial x_\alpha}{\partial X_j} - \delta_{ij} \right), \\ e_{ij} &= \frac{1}{2} \left(\delta_{ij} - \frac{\partial X_\alpha}{\partial x_i} \frac{\partial X_\alpha}{\partial x_j} \right). \end{aligned}$$

Note that E and e are symmetric. Based on this derivation, we observe that Green strain tensor measures the change of distance between two arbitrary points in the body before and after the deformation. Thus, we can conclude that Green strain tensor is invariant under rigid body motion, since during rigid body motion the distance between any points in the body stays the same.

Cauchy's infinitesimal strain tensor

If the first derivatives of the components of displacement u_i are so small that the squares and products of the partial derivatives of u_i are negligible compared with the first order terms, then e_{ij} reduces to *Cauchy's infinitesimal strain tensor*:

$$\epsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right).$$

We can see that fact by writing e in another form (in terms of displacement gradient),

$$e_{ij} = \frac{1}{2} \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} - \frac{\partial u_k}{\partial x_i} \frac{\partial u_k}{\partial x_j} \right),$$

Similarly, for E :

$$E_{ij} = \frac{1}{2} \left(\frac{\partial u_j}{\partial X_i} + \frac{\partial u_i}{\partial X_j} + \frac{\partial u_k}{\partial X_i} \frac{\partial u_k}{\partial X_j} \right),$$

Cauchy's infinitesimal strain tensor is the linear part of Almansi's strain tensor. It is worth noting that Green/Almansi strain tensor measures the deformations exactly and holds for any amount of deformation, while Cauchy's infinitesimal strain tensor only holds for small displacement, small rotation and small strain deformation.

Left and right Cauchy-Green deformation tensor

We have defined a motion mathematically by a mapping between initial and current particle positions as:

$$\vec{x} = \phi(\vec{X}, t),$$

where x denotes the current coordinate of the particle X from the original configuration. We define the *deformation gradient tensor* F as

$$F = \frac{\partial \phi}{\partial \vec{X}} = \nabla_0 \phi,$$

where $\nabla_0 \triangleq \partial/\partial X$, denotes the spatial derivatives are taken with respect to the original configuration. It is easy to check that:

$$d\vec{x} = F d\vec{X}.$$

Thus F maps the material fibers in the original or reference configuration into the corresponding material fibers in the current configuration. The deformation gradient tensor describes the deformations (rotations and stretches) of each material fiber. A useful decomposition of F is polar decomposition:

$$F = RU,$$

where R is a *rotation tensor* (thus orthogonal matrix), and U is a *stretch tensor*. Using the deformation tensor we can describe the *right Cauchy-Green deformation tensor* C , as

$$C = F^T F = U^T R^T R U = U^T U,$$

since $R^T R = I$. We see that the right Cauchy-Green deformation tensor is only composed of the stress component of the deformation tensor. Namely, C is invariant under rigid body motion. Choosing U to be a symmetric tensor, it can be uniquely determined via the relation:

$$U^2 = C.$$

Note that using the concept of stretch (ratio), one can show that C is a symmetric positive definite tensor and therefore, has real positive eigenvalues (they are called the principal stretches, see later). The left Cauchy-Green deformation tensor B is defined as

$$B = FF^T.$$

Note that a useful relation between right Cauchy-Green deformation tensor and Green strain tensor is:

$$E = \frac{1}{2}(C - I).$$

Internal virtual work

Consider a volume Ω occupied by an elastic material with the boundary $\partial\Omega$ and exterior normal n . Two types of forces can act on Ω : body force, which is described by a force density f and forces acting on the boundary $\partial\Omega$ which are assumed to have the form $\sigma \cdot n$, where σ is the Cauchy stress tensor. Summing body forces and boundary forces, we obtain that the total net force F on Ω is

$$F = \int_{\Omega} f \, dx + \int_{\partial\Omega} \sigma \cdot n \, ds.$$

Using the divergence theorem on the surface integral, we obtain

$$F = \int_{\Omega} (f + \nabla \cdot \sigma) dx.$$

We define the residual force per unit volume r as:

$$r = \nabla \cdot \sigma + f.$$

Let δu denote an arbitrary compatible virtual displacement from the current position of the body. Note that by “compatible”, we mean that the virtual displacement should satisfy the essential boundary conditions (if any). The virtual work, δW , per unit volume done by the residual force r during this virtual motion is $r \cdot \delta u$. Thus, the total virtual work of the body is:

$$\delta W = \int_{\Omega} (f + \nabla \cdot \sigma) \cdot \delta u dv.$$

Together, using the Gauss theorem, we get:

$$\nabla \cdot (\sigma \delta u) = (\nabla \cdot \sigma) \cdot \delta u + \sigma : \nabla \delta u.$$

Here we introduce the contraction operator:

$$A : B \triangleq \sum_{i,j=1}^3 A_{ij} B_{ij} = \text{tr}(A^T B),$$

for any two 3×3 matrices A and B . The internal virtual work related to stress is defined by:

$$\delta W_{int} = \int_{\Omega} \sigma : \nabla \delta u dv.$$

Further, we can decompose the matrix $\nabla \delta u$ into symmetric and anti-symmetric parts, viz.

$$A = \frac{A + A^T}{2} + \frac{(A - A^T)}{2},$$

for any matrix A . Then we can derive that:

$$\delta W_{int} = \int_{\Omega} \sigma : \delta \epsilon dv,$$

where $\delta \epsilon = \frac{1}{2}(\nabla \delta u + (\nabla \delta u)^T)$ is the Cauchy infinitesimal strain tensor correspond to the virtual displacement. Note that we have used the fact that for a symmetric matrix A (in this case σ) and an anti-symmetric matrix with zero diagonal B (the anti-symmetric part of $\nabla \delta u$):

$$A : B = 0.$$

The pair σ and ϵ are said to be *energetically conjugate* with respect to the current deformed volume in the sense that their product gives work per unit time and per current volume. If we express the virtual work equation in a material coordinate system, an alternative work conjugate pair will appear. The new pairs with respect to the original body configuration will be presented in the next section.

The second Piola-Kirchhoff stress tensor

The internal virtual work is expressed in terms of Cauchy stress whose physical meaning is force

per current configuration area. Notice that the current configuration is usually unknown (to be calculated), it would be convenient to “pull-back” the expression to the original configuration where we do know its configuration.

Let $J = \det F$, it can be shown that J gives the volume change. Namely, $dv = JdV$. The internal virtual work done by the stress is expressed as:

$$\delta W_{int} = \int_{\Omega} \sigma : \delta \epsilon \, dv = \int_{\Omega^0} \tau : \delta \epsilon \, dV,$$

where Ω^0 denotes the original configuration and $\tau = J\sigma$ is the *Kirchhoff stress tensor*. To eliminate the spatial quantities in the expression like $\delta \epsilon$, we can express the internal work alternatively as:

$$\delta W_{int} = \int_{\Omega^0} (J\sigma F^{-T}) : \delta F \, dV.$$

Define

$$P = J\sigma F^{-T},$$

as the *first Piola-Kirchhoff stress tensor*. We have omitted the derivation here, for details see [33]. A loose physical interpretation of the first Piola-Kirchhoff stress tensor is the current force per unit of undeformed area. From this equation, we find that P is energetically conjugate with F in a material coordinate system.

The 2nd Piola–Kirchhoff stress tensor S relates forces in the original configuration to areas in the original configuration via a mapping that preserves the relative relationship between the force direction and the area normal in the original configuration. It is defined as:

$$S = JF^{-1}\sigma F^{-T}.$$

If we express the internal virtual work in terms of the second Piola-Kirchhoff stress tensor, we obtain (for derivation details see [33]):

$$\delta W_{int} = \int_{\Omega^0} S : \delta E \, dV,$$

where E is the Green strain tensor. From this equation, we find that S is energetically conjugate with E in a material coordinate system.

An important property of the second Piola-Kirchhoff stress tensor is that it stays invariant under rigid body motion. This can be shown from applying a rotation (multiplying with an orthogonal matrix R) to the current configuration, then the new stress tensor S' can be written as:

$$S' = J'(RF)^{-1}(R\sigma R^T)(RF)^{-T} = JF^{-1}\sigma F^{-T} = S.$$

Note that under rigid body motion the volume of the body does not change, thus $J = J'$.

3.2.2. Material constitutive equations (linear/hyper-elasticity)

Having introduced the basics in solid mechanics, we are now ready to present the constitutive equations which define the material. We will present two main classes of material: Hookean elastic solid and hyperelastic solid.

Hookean elastic solid

A *Hookean elastic solid* is a solid that obeys Hooke's law, which states that the stress tensor is linearly proportional to the strain tensor, i.e.:

$$\sigma_{ij} = C_{ijkl}\epsilon_{kl},$$

where σ_{ij} is the Cauchy stress tensor, ϵ_{kl} is Cauchy's infinitesimal strain tensor, and C_{ijkl} is the *tensor of elastic constants*, which are independent of stress or strain.

When a material is isotropic, the array of elastic constants C_{ijkl} remains unchanged with respect to rotation and reflection of coordinates, a great reduction in the number of elastic constants is obtained. As a result, we get the isotropic Hookean solid constitutive relation as:

$$\sigma_{ij} = \lambda \epsilon_{\alpha\alpha} \delta_{ij} + 2\mu \epsilon_{ij}.$$

The constants λ and μ are called the *Lamé constants*, which are usually defined by

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

where E is Young's elastic modulus, and ν is Poisson's ratio. The physical interpretations of these two constants are: if we consider a two-dimensional problem, Young's modulus is the slope of the stress-strain curve; the Poisson's ratio is the ratio of the contraction of the surface in one dimension when one elongates it in the other. Note that Young's modulus is always a positive number and has the unit as pressure (Pascal) and Poisson ratio's dimensionless constant varies between -0.1 and 0.5 .

In engineering notation, the strain tensor ϵ can be written as a vector due to symmetry, viz.,

$$\epsilon = [\epsilon_{11}, \epsilon_{22}, \epsilon_{33}, 2\epsilon_{12}, 2\epsilon_{23}, 2\epsilon_{31}]^T.$$

Likewise, the stress tensor σ :

$$\sigma = [\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{23}, \sigma_{31}]^T.$$

Hooke's law can then be rewritten as

$$\sigma = D\epsilon,$$

where the 6×6 matrix D is given by

$$D = \begin{bmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix},$$

where the constants λ and μ are Lamé constants mentioned earlier.

Hyperelastic solid

When the work done by the stress during a deformation process is dependent only on the initial state at time t_0 and the final configuration at time t , the behavior of the material is said to be path-independent and the material is termed *hyperelastic*. Now we will present several useful aspects related to hyperelastic materials which will help us understand the hyperelastic models

that we are going to use later. For the detailed content of this section, we refer to [34].

Strain energy density function

In order to deform an elastic body, certain amount of work is required. It is stored in the body as strain energy. Due to the path-independent property of hyperelastic material, the strain energy function Ψ can be established as the work done by the stress from the initial to the current position as:

$$\Psi(F(X), X) = \int_{t_0}^t P(F(X), X) : \dot{F} dt,$$

where F is the deformation tensor, and \dot{F} , its derivative with respect to time and P is the first Piola-Kirchhoff stress tensor. Note that we have used the fact that P and F are energetically conjugate.

Since the elastic potential must remain invariant under a rigid body rotation, Ψ should depend on F only via the stretch component U and independent of R . Hence, Ψ is usually expressed as a function of $C = U^2 = F^T F$ as:

$$\Psi(F(X), X) = \Psi(C(X), X).$$

Thus the constitutive relations can be obtained by observing that

$$\dot{\Psi} = \frac{\partial \Psi}{\partial C} : \dot{C} = \frac{1}{2} S : \dot{C},$$

$$\frac{1}{2} \dot{C} = \dot{E},$$

$$S(C(X), X) = 2 \frac{\partial \Psi}{\partial C} = \frac{\partial \Psi}{\partial E},$$

where S is the second Piola-Kirchhoff stress tensor and C is the right Cauchy deformation tensor and E is the Green strain tensor. This equation is often used as a definition of a hyperelastic material. In other words, the constitutive equation of a hyperelastic material is determined by the strain energy function.

Isotropic material

Because of the isotropic property, we expect that the relationship between Ψ and C must be independent of the material axes. Therefore, Ψ should only be a function of the invariants of C , as:

$$\Psi(C(X), X) = \Psi(I_C, II_C, III_C, X),$$

where I_C , II_C and III_C are the invariants of C . They are intrinsic magnitudes and remain invariant under the rotation of axes. They are defined as:

$$I_C = \text{tr } C = \sum_{i=1}^3 C_{ii},$$

$$II_C = C : C = \text{tr}(C^T C),$$

$$III_C = \det C.$$

Their relation with the principal stretches (the eigenvalues of C) $\lambda_1, \lambda_2, \lambda_3$ is:

$$I_C = \lambda_1 + \lambda_2 + \lambda_3,$$

$$II_C = \lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_2\lambda_3,$$

$$III_C = \lambda_1\lambda_2\lambda_3.$$

After some computation, the second Piola-Kirchhoff stress can be evaluated in terms of invariants of C as:

$$S = 2\Psi_I + 4\Psi_{II}C + 2J^2\Psi_{III}C^{-1},$$

$$\text{where } \Psi_I = \frac{\partial\Psi}{\partial I_C}, \Psi_{II} = \frac{\partial\Psi}{\partial II_C}, \Psi_{III} = \frac{\partial\Psi}{\partial III_C}.$$

Material models and uniaxial tension test

Below we introduce five hyperelastic material models which have been used (or suggested to be used) in soft tissue modelling over the years. Then, a figure showing stress-strain relationship under uniaxial test of different material models is given as an illustration of the difference between the material models. Note that except from the St.Venant-Kirchhoff model, the rest of the strain density functions are given for the case when the material is totally incompressible. Thus, the strain density function is stated with the implicit constraint $III_C = 1$.

1. St.Venant-Kirchhoff model

The simplest example of a hyperelastic material is the St.Venant-Kirchhoff model, which is defined by a strain energy density function Ψ as:

$$\Psi(E) = \frac{1}{2}\lambda(trE)^2 + \mu E:E,$$

where λ and μ are Lamé constants and E is Green strain tensor. We can obtain the second Piola-Kirchhoff stress tensor as

$$S = \lambda(trE)I + 2\mu E.$$

Note that the last equation is analogous to the constitutive relation in linear elasticity, where Cauchy's infinitesimal strain tensor has been replaced by the Green strain tensor and Cauchy stress has been replaced by the second Piola-Kirchhoff stress tensor. Although St.Venant-Kirchhoff material has been found to be of little practical use beyond the small strain regime, it takes the geometric nonlinearity into account and is thus suitable for large displacement, large rotation but small strain problems. This will be explained with more details in Section 4.1.3.

2. Mooney-Rivlin model

A general form for the strain energy density function of incompressible rubbers attributable to Mooney and Rivlin is expressed as:

$$\Psi(C) = \sum_{r,s \geq 0} C_{rs}(I_C - 3)^r(II_C - 3)^s,$$

where I_C and II_C are the first and the second invariant of C respectively. The most frequently used from this family of equations is obtained when only C_{01} and C_{10} are different from zero. In this particular case we have:

$$\Psi(C) = C_{10}(I_C - 3) + C_{01}(II_C - 3).$$

Since $\mu (= 2(C_{10} + C_{01}))$ denotes the shear modulus (non-negative), thus we need

$$C_{10} + C_{01} \geq 0.$$

3. Neo-Hookean model

With $C_{01} = 0$, Neo-Hookean material is defined by a hyperelastic strain energy density function Ψ given as:

$$\Psi = C_1(I_C - 3).$$

In this case, $\mu = 2C_1 \geq 0$, thus $C_1 \geq 0$.

4. Yeoh model

The strain energy density function of the Yeoh model is written as:

$$\Psi = \sum_{i=1,2,3} C_i(I_C - 3)^i,$$

where C_i are material constants. The compatibility condition for the Yeoh model is that $2C_1 = \mu$. Thus $C_1 \geq 0$.

5. Gent model

In this model, the strain energy density function is designed such that it has a singularity when the first invariant of the left Cauchy-Green deformation tensor reaches a limiting value I_m . The strain energy density function for the Gent model is:

$$\Psi = -\frac{\mu J_m}{2} \ln \left(1 - \frac{I_C - 3}{J_m} \right),$$

where μ is the shear modulus and $J_m = I_m - 3$. In the limit ($I_m \rightarrow \infty$), the Gent model reduces to the Neo-Hookean solid model.

Note that for incompressible material, the independent variables in terms of invariants of C are I_C and II_C . I_C and II_C satisfy:

$$\begin{aligned} I_C &\geq 3, \\ II_C &\geq 3. \end{aligned}$$

This is derived by calculating the minima of first and second invariants of C under the constraint $III_C = 1$. The calculation goes as follows:

Since

$$III_C = \lambda_1 \lambda_2 \lambda_3 = 1,$$

then

$$I_C = \lambda_1 + \lambda_2 + \lambda_3 = \frac{1}{\lambda_2 \lambda_3} + \lambda_2 + \lambda_3.$$

To calculate the minima of I_C , we need to calculate the root of partial derivatives of I_C with respect to λ_2 and λ_3 . We have:

$$\frac{\partial I_C}{\partial \lambda_2} = 1 - \frac{1}{\lambda_3} \frac{1}{\lambda_2^2} = 0,$$

and

$$\frac{\partial I_C}{\partial \lambda_3} = 1 - \frac{1}{\lambda_2} \frac{1}{\lambda_3^2} = 0.$$

We have pointed out before that the eigenvalues of C are all real and positive. Solving above equations, we found that I_C reaches its minima when $\lambda_2 = \lambda_3 = 1$. This also implies $\lambda_1 = 1$. Thus we found that

$$I_C - 3 = (\lambda_1 - 1) + (\lambda_2 - 1) + (\lambda_3 - 1) \geq 0.$$

In a similar way, we have:

$$II_C - 3 = (\lambda_1 \lambda_2 - 1) + (\lambda_1 \lambda_3 - 1) + (\lambda_2 \lambda_3 - 1) \geq 0.$$

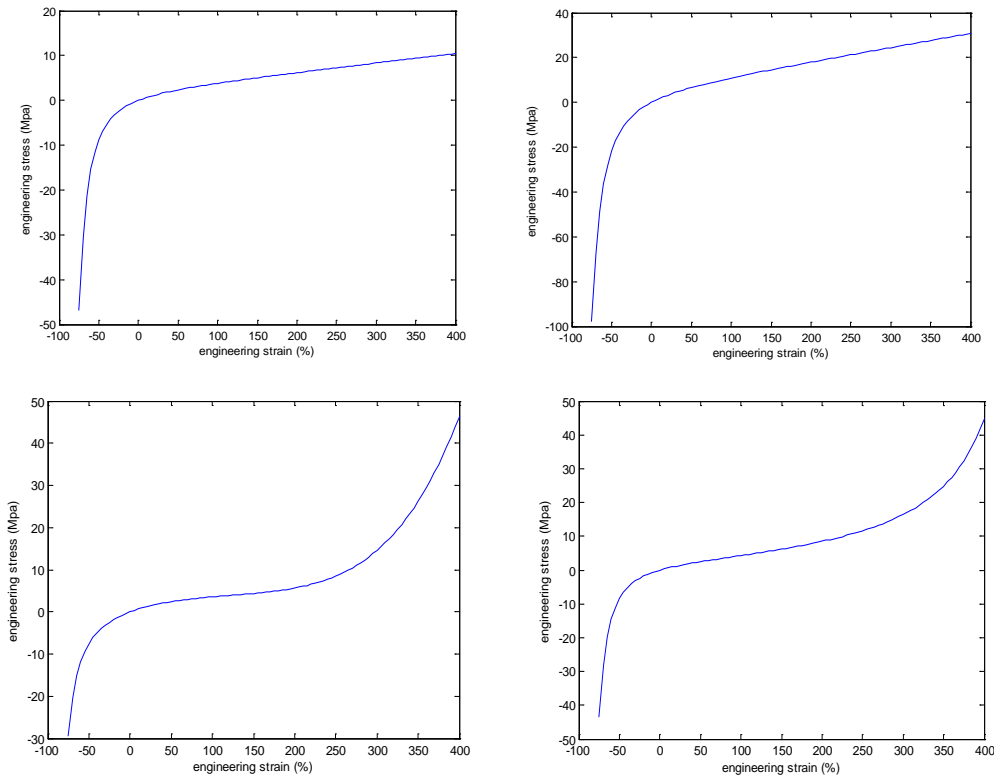
And when $F = I$ (identity matrix), $\lambda_i = 1$, thus $I_C - 3 = II_C - 3 = 0$.

A consistent strain energy function should satisfy the following criteria (and many more, see [44]):

1. The strain energy density function must be non-negative for all deformations.
2. The strain energy density function must have a zero value at the undeformed state.

The calculation of I_C and II_C shows that the presented material models are consistent under a mild constraint on the parameters (like in the Yeoh model).

It is common to use a uniaxial test procedure to test the mechanical properties of a certain material. With uniaxial testing, length of the material is deformed in one direction and the resulting stress-strain relationship determined. Below we are going to present the uniaxial test for the material mentioned above (except from St.Venant-Kirchhoff material), with specified parameters.



Note that all figures are drawn under the x -axis “engineering strain” (ranging from $-75\% \sim 400\%$) and y -axis “engineering stress” with the unit MPa. These figures are plotted using the calculated strain-stress relation in terms of the principal stretch. Below is type of material and the material constants used in each figure:

- Up-left: Mooney-Rivlin, $C_{10} = 1.030\text{MPa}$, $C_{01} = 0.114\text{MPa}$.
- Up-right: Neo-Hookean, $C_1 = 3.1\text{MPa}$.
- Bottom-left: Yeoh, $C_1 = 1.202\text{MPa}$, $C_2 = -0.057\text{MPa}$ and $C_3 = 0.004\text{MPa}$.
- Bottom-right: Gent, $\mu = 2.290\text{MPa}$, $J_m = 30$.

By plotting the stress-strain relationship for different hyperelastic materials, we are making the effort to find the best match for human skin stress-strain relationship shown in Figure 3.5. By comparing the figures, we find that Yeoh and Gent model has more similarity with the target shape than Mooney and Neo-Hookean material.

Incompressible material and near incompressibility

Incompressible media have the property that after deformation each small portion of the medium has the same volume as before deformation. Soft tissues due to the high percentage of water content are usually treated as quasi-incompressible. Since for incompressible material the constraint $\det F = J = 1$ must be fulfilled, special care must be taken when we deal with this type of problem. For instance, the constitutive equation of isotropic compressible Hookean solids reads:

$$\sigma_{ij} = \lambda \epsilon_{\alpha\alpha} \delta_{ij} + 2\mu \epsilon_{ij},$$

where the constants are determined by,

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

where E is Young's modulus and ν is Poisson's ratio. It can be shown that $\nu = \frac{1}{2}$ represents incompressibility. Thus the parameter λ will become unbounded in the incompressible limit. Therefore, an alternative formulation of the theory is needed.

Taking Mooney-Rivlin material as an example, the strain energy density function for totally incompressible material reads:

$$\Psi(C) = C_{10}(I_C - 3) + C_{01}(II_C - 3),$$

with the constraint

$$III_C = 1.$$

To drop the restriction $III_C = 1$, we include a hydrostatic work term in the strain energy function to obtain

$$\Psi'(C) = C_{10}(I_C - 3) + C_{01}(II_C - 3) + W(III_C).$$

However, this expression is not convenient from a computational point of view. Thus the final strain density function used in computation is written as:

$$\Psi''(C) = \widehat{\Psi}'(C) + U(III_C),$$

where $\widehat{\Psi}'(C) = \Psi'\left(III_C^{-\frac{1}{3}}C\right)$ is the deviatoric part of the strain density function Ψ and $U(III_C)$ is the volumetric part. Material which satisfies such a strain density form is called near incompressible. The terminology “near incompressibility” is used to denote materials that are truly/almost incompressible, but the numerical treatment invokes a small measure of volumetric deformation. Thus, near incompressibility is often a device by which incompressibility can more readily be enforced within the context of the finite element formulation [34]. More details on the formulation within the finite element context will be given in Section 4.1.4.

4. Mathematical models and implementation details

With the background knowledge presented in the previous section, we are now ready to introduce the facial soft tissue models we developed in this project. As mentioned in section 2.2, FEM has superior performance over other methods in terms of accuracy. Due to the high demand of accuracy and the availability of the FEM software package Marc, we decide to mainly focus on the development of FEM-based soft tissue models in this project. In this chapter, we will give a detailed mathematical description of the FEM-based models and a computer-vision based method which is also used for mouth-opening. This chapter is organized as follows: First, we describe our FEM-based facial soft tissue models in the following aspects: general model assumptions, the configuration of the models and governing equations for the systems. Then, the FEM approach which we deployed to solve the problem is presented in great details. It is followed by a brief description of the morphing method. At the end of the chapter, a summary (table) of the methods we used for the mouth-opening tool and a step-by-step computer implementation procedure are stated.

4.1. FEM-based facial soft tissue models

In this section, we will present our FEM-based models in the following fashion: to start with, we list the assumptions we made at the modelling stage. Then we illustrate the configuration of our models, from which one can get an intuitive idea about the construction and the structure of our mesh. What follows are the governing equations that describe the physical system mathematically. After transforming the problem into mathematical equations, we apply the numerical solver (FEM) to solve the equations.

4.1.1. General model assumptions

The following simplifications are made in all the FEM-based models:

1. **Plasticity ignored**

It is only necessary to model the strain and stress range that is relevant to the open mouth process. The part of the stress-strain curves which represents the very large strain is rarely invoked by the open mouth process.

2. **Viscoelasticity ignored**

Viscoelasticity will only play a role in dynamic modelling. Since we are only interested in the static result, namely the open mouth geometry, the viscoelastic property can be ignored.

3. **Initial stress free**

For the initial state of the model, the skin is considered with zero tension in the resting state (stress free). Note that this is not completely true as there exists a pretension in the skin [30].

4. **Isotropy and homogeneity**

All the soft tissue layers in the models are considered to be isotropic and homogeneous. From Section 3.1.2, we know this assumption is not realistic but a compromise to gain computational simplicity. But notice that the heterogeneity is partly taken care of by adopting a layered structure of soft tissues.

5. **Gravity ignored**

Although human faces suffer gravity, we ignore body force (gravity) in all the models, since we deem the effect of the gravity on the facial geometry is small compared to deformation due to the mouth opening process.

6. No muscles modelled

Another significant simplification of the models is ignoring the existence of the muscles and the forces caused by muscles in the mouth-opening process.

4.1.2. Configuration of the model

The model configuration mainly comprises of a two-layer structure (skin layer and skull layer) with connecting springs in between. What we are interested in is the displacement field of the skin surface layer. We set the skull layer as rigid to provide the movement of the jaw. The two layers are connected by springs, which are designed to model the sliding effect of the superficial fascia. Further, the skull layer is divided into two parts by a “cut” along the x -axis at the mouth height: lower skull and upper skull. The lower skull can move freely in 3D and therefore simulate the moving jaw in the mouth opening process. Figure 4.1 is a snapshot from MENTAT (GUI of Marc) illustrating the configuration of the models.

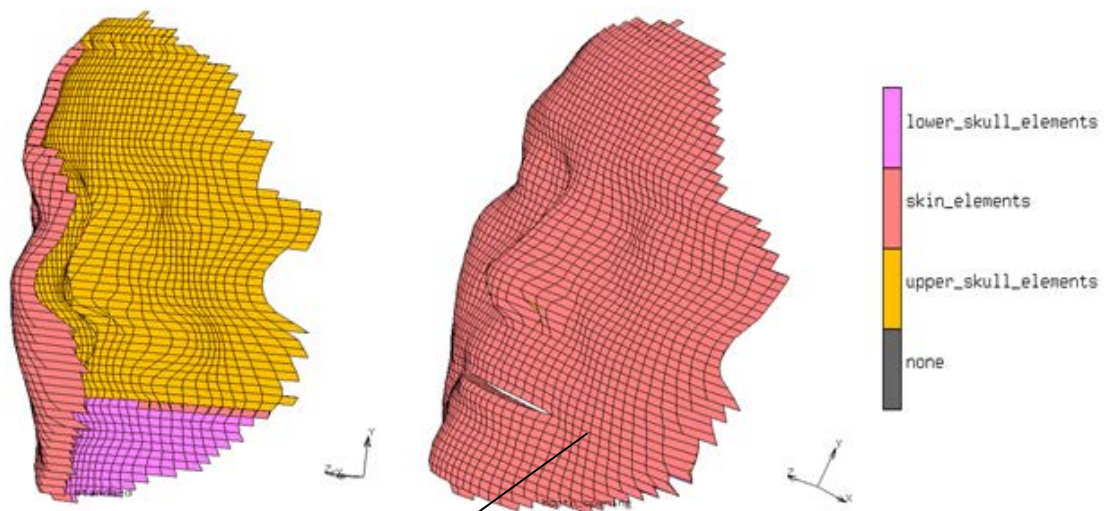


Figure 4.1 Layer structure illustration

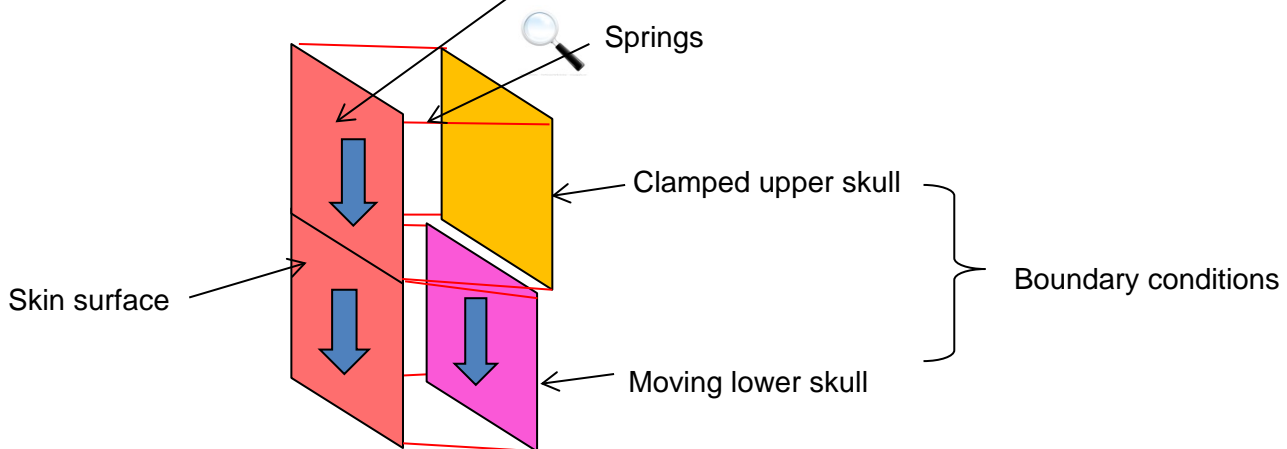


Figure 4.2 Local element illustration

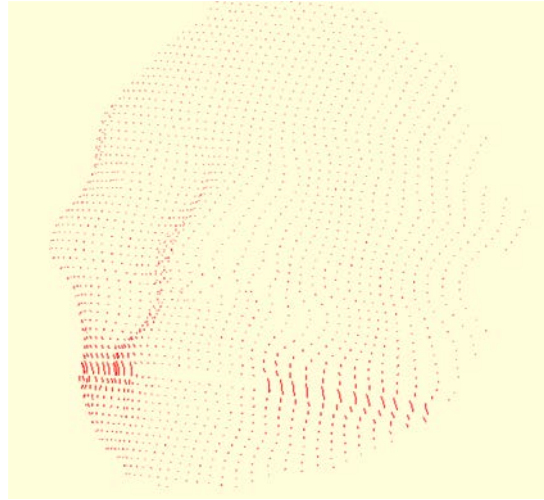


Figure 4.3 springs elongation during mouth opening

It is worth to point out that to “create” a mouth in the model and a free moving jaw, another two sets of nodal points are added at the mouth position at the skin layer and at the upper-lower skull border line. These nodes are assigned to have the same x, y -coordinates with the original nodes but 0.5 mm projection (outwards) in the z -direction. Therefore, after some modification on the connectivity of the elements, the original nodes at the skin layer represent the position of the upper lip and the added nodes the lower lip.

In the local elements illustration, we magnified four elements which are located at the side of the mouth. Note that all skin nodes have a one to one mapping to skull nodes except from the ones on the line containing the mouth opening. On the sides of the mouth every skin node has two spring connections: to the upper skull and to the jaw. We recognized that this is caused by the addition of nodes: in the skin layer we only added the nodes in the lip position with the width of a mouth, while in the skull layer we added the nodes across the head width. From the illustration, one can also observe that we use uniform grids for both layers.

4.1.3. Governing equations

Having specified the model assumptions and the configuration, we are ready to present the governing equations of the system. Mathematically speaking, the governing equations of the system consist of physical equilibrium laws, constitutive equations, and boundary conditions. Below we will elaborate each of the items to provide a full-round mathematical description of the system.

Cauchy’s equilibrium equation

The equilibrium law in solid mechanics is attributed to Cauchy’s force equilibrium equation. Let Ω denote our domain of interest – the skin layer – and let ω denote an arbitrary subdomain of Ω with the boundary $\partial\omega$ and exterior normal n . Two types of forces can act on ω , namely, body forces and surface traction forces. Since we ignore gravity in the models, we only consider surface traction force, which, when it acts on the boundary $\partial\omega$, takes the form $\sigma \cdot n$, where σ is the Cauchy stress tensor introduced in section 3.2.1. Summing up the total net force F on Ω , we get

$$F = \int_{\partial\omega} \sigma \cdot n \, ds.$$

Using the divergence theorem on the surface integral, we obtain

$$F = \int_{\omega} (\nabla \cdot \sigma) dx.$$

In equilibrium, $F = 0$. Since ω is arbitrary, we conclude that

$$\nabla \cdot \sigma = 0, \quad \text{in } \Omega.$$

Hereby we have derived Cauchy's equilibrium equation.

Boundary conditions (postulations)

We point out very strongly that we have used Marc to assign boundary conditions by manipulating the movement of lower and upper skull, but how exactly Marc treats spring support is not yet completely known (further study is needed). Thus what is stated below is my personal understanding of the boundary conditions which is applied to the skin layer. To understand what happens exactly in Marc, it requires more structural mechanics knowledge within the finite element context (where they treat springs as a special finite element as well) and more familiarity with Marc as well. It is out of the scope of this thesis.

To obtain a unique solution u , Cauchy's equilibrium equation must be supplemented by suitable boundary conditions. According to the configuration of the model, the boundary conditions should be induced by the spring forces which are applied to every node at the inner side of skin layer, Γ_{in} . There are concentrated surface tractions applied on Γ_{in} . In mathematical terms, the boundary conditions are of Neumann type, which take the form

$$\sigma \cdot n = g, \quad \text{on } \Gamma_{in},$$

where σ is the Cauchy stress tensor and n is the outward unit normal to the surface, and g is traction.

Basically, we model the open-mouth process as prescribed displacement of the lower skull (the moving jaw) and the clamped upper skull. Through the rotation and translation of the lower skull (just like the movement of the jaw when people open their mouth), springs connecting lower skull nodes and corresponding skin nodes will produce pulling forces on the inner side of skin layer. At the same time, since the upper skull is clamped, any displacement of the skin layer nodes which are connected to the upper skull nodes, will lead to the elongation of the springs as well. An illustration of the elongation of the spring described above is Figure 4.3, where we can observe the elongation of the springs both in the lower and upper skull. The stretching effect is the most obvious at the side of the mouth, which is realistic.

As pointed out in section 4.1.2, the skin nodes located at the sides of mouth are connected with two springs (one with lower skull nodes and one with upper skull nodes) and the rest of the skin nodes have a one to one connection with the skull nodes. Thus, it is sensible to group skin nodes when we discuss boundary conditions.

- I. Skin nodes that have only one spring connection. Further we divide this group into two subgroups:
 - a) Nodes that are connected to lower skull nodes.
In this case, the surface traction is caused by the rigid movement of the lower skull. We denote the displacement vector (constant vector) of the skull nodes as D . Thus the surface traction g_{II_a} is:

$$g_{II_a}(x) = \mathcal{K}(D - u(x))\delta(x - x_k), \quad x \in \Gamma_{in}, \quad k \in N_{II_a},$$

where \mathcal{K} is the spring stiffness, N_{II_a} is the index set of nodes in this category, and x_k denotes the coordinates in the Cartesian coordinate system of node k , $u(x)$ denotes the displacement field, defined as $x - X$, for every point in the body. Note that $\delta(x)$ is the delta function, which has the property:

$$\int_{\Omega} w(x) \delta(x - y) d\Omega = w(y),$$

for any continuous function $w(x)$ defined on Ω . We use the delta function to represent a concentrated force which in this case is the force applied on the discretized nodes on the surface.

- b) Nodes that are connected to upper skull nodes. We can write the surface traction g_{II_b} in a similar way like g_{II_a} . The difference is that for the nodes connected to the clamped upper skull, the stretching/contraction of the springs comes from the displacement of skin nodes only. We have

$$g_{II_b} = \mathcal{K}(0 - u(x))\delta(x - x_k), \quad x \in \Gamma_{in}, \quad k \in N_{II_b},$$

where N_{II_b} denotes the index set of nodes in this category.

- II. Skin nodes that have two springs connections. For this type of skin nodes, the total traction g_I should be the sum of traction that is induced by the springs connecting both the lower and upper skull nodes, g_l and g_u , namely:

$$g_I = g_l + g_u = \mathcal{K}[(D - u(x))\delta(x - x_k) + (0 - u(x))\delta(x - x_k)], \quad k \in N_I,$$

where N_I is the index set of the nodes belonging to this category.

Thus to summarize, the boundary conditions of the mouth opening process can be written as:

$$\sigma \cdot n = g, \quad \text{on } \Gamma_{in},$$

where

$$g = g_I + g_{II_b} + g_{II_a}, \quad \text{on } \Gamma_{in}.$$

Principle of virtual work

The principle of virtual work is of utmost importance in the engineering world and it is also the corresponding weak form of the Cauchy equilibrium equation. Below we first derive the weak form of the Cauchy equilibrium equation and then we give its engineering interpretation, namely, as the principle of virtual work. At the end, we will present total and updated Lagrangian formulations which are used to linearize the nonlinear problems in the solution procedure.

Weak form derivation

Let V be the Hilbert space:

$$V = \{v \in [H^1(\Omega)]^3 \mid v = 0 \text{ on } \Gamma \setminus \Gamma_{in}\}.$$

Multiplying $\nabla \cdot \sigma = 0$ with a test function $v \in V$, and next integrating by parts, we get

$$0 = \int_{\Omega} (\nabla \cdot \sigma) v \, dx = \int_{\Omega} \sum_{i,j=1}^3 \left(\frac{\partial \sigma_{ij}}{\partial x_j} v_i \right) dx$$

$$= \int_{\partial\Omega} \left(\sum_{i,j=1}^3 \sigma_{ij} n_j v_i \right) dS - \int_{\Omega} \sum_{i,j=1}^3 \left(\sigma_{ij} \frac{\partial v_i}{\partial x_j} \right) dx.$$

Thus, we have:

$$\int_{\partial\Omega} ((\sigma \cdot n)v) dS - \int_{\Omega} (\sigma : \nabla v) dx = 0.$$

Using finally the Neumann boundary condition $\sigma \cdot n = g$ on Γ_{in} , and that $v = 0$ on $\Gamma \setminus \Gamma_{in}$, we end up with

$$\int_{\Omega} (\sigma : \nabla v) dx = \int_{\Gamma_{in}} (gv) dS, \quad \forall v \in V.$$

We can further simplify the result. Any matrix can be decomposed into its symmetric and anti-symmetric part, viz., $A = \frac{A+A^T}{2} + \frac{(A-A^T)}{2}$. Thus, it follows that

$$\sigma : \nabla v = \sigma : \frac{1}{2}(\nabla v + \nabla v^T) + \sigma : \frac{1}{2}(\nabla v - \nabla v^T) = \sigma : \epsilon(v) + 0,$$

where $\epsilon(v) = \frac{1}{2}(\nabla v + \nabla v^T)$. So we get

$$\int_{\Omega} (\sigma : \epsilon(v)) dx = \int_{\Gamma_{in}} (gv) dS, \quad \forall v \in V.$$

Or we can write it in another notation:

$$\int_{\Omega} \sigma_{ij} \epsilon(v)_{ij} dx = \int_{\Gamma_{in}} (g_i v_i) dS, \quad \forall v \in V.$$

Note that the double appearance of the index means summing over the spatial dimension, which is three in this case. If we interpret v as virtual displacement, then $\epsilon(v)$ is the corresponding infinitesimal virtual strain and σ_{ij} is the real Cauchy stress. The right hand side can be seen as the external virtual work. We rewrite the above equation in engineering notation as:

$$\int_{\Omega} \sigma_{ij} \delta \epsilon_{ij} dx = \int_{\Gamma_{in}} (g_i \delta u_i) dS, \quad \forall \delta u \in V.$$

Note that the δ in front the variable denotes a virtual variable.

Total Lagrangian formulation (T.L. formulation)

To solve a nonlinear problem, a linearization procedure is necessary and usually an incremental solution is taken. We select Newton-Raphson iteration solution procedure. Although it is theoretically possible to achieve a direct solution for a given load case, it is more practical to consider the external load as being applied as a sum of increments. We apply the external load evenly in 50 time steps. For details of the content of the following three sections (Total Lagrangian formulation (T.L. formulation), Updated Lagrangian formulation (U.L. formulation), and Lagrangian elasticity tensor) we refer to [36].

Lagrangian description is suitable for a body subjected to large displacement, large rotation and large strain. Assume that we have obtained the solution (the displacement field) till time t . And

we would like to have the solution at time $t + \Delta t$ (the next time step). Let u denote the displacement increment, namely:

$$u^{t+\Delta t} = u^t + u.$$

To solve the problem, we apply the principle of virtual work at time $t + \Delta t$:

$$\int_{\Omega^{t+\Delta t}} \sigma_{ij}^{t+\Delta t} \delta \epsilon_{ij}^{t+\Delta t} dx = \int_{\Gamma_{in}^{t+\Delta t}} ((g^{t+\Delta t})_i \delta u_i) dS = \mathcal{R}^{t+\Delta t}, \quad \forall \delta u \in V,$$

where $\mathcal{R}^{t+\Delta t}$ denotes the external force term. Note that the superscript $t + \Delta t$ is to indicate the variable at time $t + \Delta t$. Using the property of the δ -function, we can write out $\mathcal{R}^{t+\Delta t}$ explicitly:

$$\mathcal{R}^{t+\Delta t} = \int_{\Gamma_{in}^{t+\Delta t}} ((g^{t+\Delta t})_i \delta u_i) dS = \sum_{k \in N} {}^k R_{t+\Delta t} \delta u_i, \quad i = 1, 2, 3,$$

where N denotes the whole nodes set on $\Gamma_{in}^{t+\Delta t}$, namely, $N = N_I \cup N_{II_b} \cup N_{II_a}$. ${}^k R_{t+\Delta t}$ denotes the force applied on the k th node by the springs in the x_i direction at time $t + \Delta t$ as:

$$\begin{aligned} {}^k R_{t+\Delta t} &= \mathcal{K}(D_i^{t+\Delta t} - u_i^{t+\Delta t}), & \text{if } k \in N_{II_a}, \\ {}^k R_{t+\Delta t} &= \mathcal{K}(0 - u_i^{t+\Delta t}), & \text{if } k \in N_{II_b}, \\ {}^k R_{t+\Delta t} &= \mathcal{K}[(D_i^{t+\Delta t} - u_i^{t+\Delta t}) + (0 - u_i^{t+\Delta t})], & \text{if } k \in N_I, \end{aligned}$$

where $U_i^{t+\Delta t}$ denotes the i th component of the displacement vector U at time step $t + \Delta t$. We use a uniform time step. If the time $t + \Delta t$ is the p th step out of the 50, we have

$$D_i^{t+\Delta t} = \left(p \frac{D}{50}\right)_i.$$

Note that the external force term is deformation dependent. Thus approximation is needed, and the treatment will be given in section 4.1.4.

In section 3.2.1, we presented that the second Piola-Kirchhoff stress tensor is energetically conjugate with the Green strain tensor. They are the counterparts of the Cauchy stress tensor and the infinitesimal strain tensor at the original configuration ($t = 0$) respectively. Using this fact, we can rewrite the principle of virtual work at time $t + \Delta t$ referred to the original configuration as

$$\int_{\Omega^0} S_{ij}^{t+\Delta t} \delta E_{ij}^{t+\Delta t} dx = \mathcal{R}^{t+\Delta t}, \quad \forall \delta u \in V,$$

where $S_{ij}^{t+\Delta t}$ denotes the second Piola-Kirchhoff stress (implicitly) referred to the original configuration, namely

$$S_{ij}^{t+\Delta t} = J^{t+\Delta t} F^{-1(t+\Delta t)} \left(x^{t+\Delta t}(X) \right) \sigma_{ij}^{t+\Delta t} \left(x^{t+\Delta t}(X) \right) F^{t+\Delta t}(X),$$

where $F_{ij}^{t+\Delta t}(X) = \frac{\partial x_i^{t+\Delta t}}{\partial X_j}$, $J^{t+\Delta t} = \det(F^{t+\Delta t}(X))$, X denote the original configuration of the domain, namely, the skin layer at time $t = 0$. The Green strain tensor is rewritten with respect to the original configuration as:

$$E_{ij}^{t+\Delta t} = \frac{1}{2} \left(\frac{\partial u_j^{t+\Delta t}}{\partial X_i} + \frac{\partial u_i^{t+\Delta t}}{\partial X_j} + \frac{\partial u_k^{t+\Delta t}}{\partial X_i} \frac{\partial u_k^{t+\Delta t}}{\partial X_j} \right).$$

Further, we decompose the unknown stress and strain (at time $t + \Delta t$) as:

$$S_{ij}^{t+\Delta t} = S_{ij}^t + S_{ij},$$

$$E_{ij}^{t+\Delta t} = E_{ij}^t + E_{ij}.$$

Thus, S_{ij}^t and E_{ij}^t are the known parts since they are the variables at time t . S_{ij} and E_{ij} are the unknown increments. Next, let us expand the incremental Green strain tensor expression. We get

$$\begin{aligned} E_{ij} = E_{ij}^{t+\Delta t} - E_{ij}^t &= \frac{1}{2} \left(\frac{\partial (u_j^{t+\Delta t} - u_j^t)}{\partial X_i} + \frac{\partial (u_i^{t+\Delta t} - u_i^t)}{\partial X_j} + \frac{\partial u_k^{t+\Delta t}}{\partial X_i} \frac{\partial u_k^{t+\Delta t}}{\partial X_j} - \frac{\partial u_k^t}{\partial X_i} \frac{\partial u_k^t}{\partial X_j} \right) \\ &= \frac{1}{2} \left(\frac{\partial u_j}{\partial X_i} + \frac{\partial u_i}{\partial X_j} + \frac{\partial u_k^t}{\partial X_i} \frac{\partial u_k}{\partial X_j} + \frac{\partial u_k}{\partial X_i} \frac{\partial u_k^t}{\partial X_j} \right) + \frac{1}{2} \left(\frac{\partial u_k}{\partial X_i} \frac{\partial u_k}{\partial X_j} \right). \end{aligned}$$

Note that the first term is linear with respect to the unknown displacement increments u , while the second term is nonlinear in u . We note $\delta E_{ij}^{t+\Delta t} = \delta E_{ij}$, since each variation is taken on the displacements at time $t + \Delta t$, with u^t fixed. Let

$$\theta_{ij} = \frac{1}{2} \left(\frac{\partial u_j}{\partial X_i} + \frac{\partial u_i}{\partial X_j} + \frac{\partial u_k^t}{\partial X_i} \frac{\partial u_k}{\partial X_j} + \frac{\partial u_k}{\partial X_i} \frac{\partial u_k^t}{\partial X_j} \right)$$

denote the linear part of the strain increment and

$$\eta_{ij} = \frac{1}{2} \left(\frac{\partial u_k}{\partial X_i} \frac{\partial u_k}{\partial X_j} \right)$$

is the nonlinear strain increment. Thus $E_{ij} = \eta_{ij} + \theta_{ij}$. Substitute the expressions of the unknown $S_{ij}^{t+\Delta t}$ and $E_{ij}^{t+\Delta t}$ into the principle of virtual work formulation to get:

$$\int_{\Omega^0} S_{ij}^{t+\Delta t} \delta E_{ij} dx = \int_{\Omega^0} (S_{ij}^t + S_{ij}) (\delta \theta_{ij} + \delta \eta_{ij}) dx = \mathcal{R}^{t+\Delta t}.$$

Rewrite it as:

$$\int_{\Omega^0} S_{ij} \delta E_{ij} dx + \int_{\Omega^0} S_{ij}^t \delta \eta_{ij} dx = \mathcal{R}^{t+\Delta t} - \int_{\Omega^0} S_{ij}^t \delta \theta_{ij} dx.$$

Notice that the right hand side is known for any given variation δu and no approximation has been applied yet. For hyperelastic material, the term S_{ij} is usually nonlinear in u . Now we are going to linearize this term as a necessary step in solving a nonlinear problem. Note that

$$\delta \eta_{ij} = \frac{1}{2} \left(\left(\delta \frac{\partial u_k}{\partial X_i} \right) \frac{\partial u_k}{\partial X_j} + \frac{\partial u_k}{\partial X_i} \left(\delta \frac{\partial u_k}{\partial X_j} \right) \right),$$

which is linear in u . Therefore, the second term at the left hand side is also linear in the unknown u . Now we perform the linearization procedure on the first term. Firstly, we expand $S_{ij}^{t+\Delta t}$ as a Taylor series at E_{ij}^t . Let us denote the mapping between strain and stress tensor as f . Namely, $f(E_{ij}^t) = S_{ij}^t$, $\forall i, j = 1, 2, 3$. Then

$$S_{ij}^{t+\Delta t} = f(E_{ij}^{t+\Delta t}) = f(E_{ij}^t) + (E_{ij}^{t+\Delta t} - E_{ij}^t) \frac{\partial f}{\partial E}(E_{ij}^t) + \text{higher order terms.}$$

We get:

$$S_{ij}^{t+\Delta t} - S_{ij}^t = S_{ij} = E_{rs} \frac{\partial S_{ij}^t}{\partial E_{rs}^t} + \text{higher order terms} \cong E_{rs} \frac{\partial S_{ij}^t}{\partial E_{rs}^t} = (\theta_{rs} + \eta_{rs}) \frac{\partial S_{ij}^t}{\partial E_{rs}^t} \cong \theta_{rs} \frac{\partial S_{ij}^t}{\partial E_{rs}^t}.$$

Note that we make the approximation first by ignoring the higher order terms in S_{ij} , then we also ignore the nonlinear part of the incremental strain. Hence, we obtain

$$S_{ij} \delta E_{ij} \cong \theta_{rs} \frac{\partial S_{ij}^t}{\partial E_{rs}^t} (\delta \theta_{ij} + \delta \eta_{ij}) = \theta_{rs} \frac{\partial S_{ij}^t}{\partial E_{rs}^t} \delta \theta_{ij} + \theta_{rs} \frac{\partial S_{ij}^t}{\partial E_{rs}^t} \delta \eta_{ij} \cong \theta_{rs} \frac{\partial S_{ij}^t}{\partial E_{rs}^t} \delta \theta_{ij}.$$

Note that we throw away the term $\theta_{rs} \frac{\partial S_{ij}^t}{\partial E_{rs}^t} \delta \eta_{ij}$ in the last step approximation because $\theta_{rs} \frac{\partial S_{ij}^t}{\partial E_{rs}^t} \delta \eta_{ij}$ is nonlinear in u (two linear terms multiplication). Thus the final linearized equation is:

$$\int_{\Omega^0} \frac{\partial S_{ij}^t}{\partial E_{rs}^t} \theta_{rs} \delta \theta_{ij} dx + \int_{\Omega^0} S_{ij}^t \delta \eta_{ij} dx = \mathcal{R}^{t+\Delta t} - \int_{\Omega^0} S_{ij}^t \delta \theta_{ij} dx.$$

Updated Lagrangian formulation (U.L. formulation)

In the last section, we present the total Lagrangian formulation in which all variables are taken reference to the original configuration. In this section we are going to present another formulation which is also used in our solution procedure. The updated Lagrangian formulation takes the reference configuration at time t when we solve the principle of virtual work at time $t + \Delta t$.

Just like in the total Lagrangian case, we are going to write the principle of virtual work at time $t + \Delta t$ with time reference t as

$$\int_{\Omega^t} S_{ij}^{t+\Delta t} \delta E_{ij}^{t+\Delta t} dx = \mathcal{R}^{t+\Delta t}, \quad \forall \delta u \in V,$$

where

$$S_{ij}^{t+\Delta t} = J^{t+\Delta t} F^{-1(t+\Delta t)}(x^{t+\Delta t}(x^t)) \sigma_{ij}^{t+\Delta t}(x^{t+\Delta t}(x^t)) F^{t+\Delta t}(x^t),$$

and

$$E_{ij}^{t+\Delta t} = \frac{1}{2} \left(\frac{\partial u_j^{t+\Delta t}}{\partial x_i^t} + \frac{\partial u_i^{t+\Delta t}}{\partial x_j^t} + \frac{\partial u_k^{t+\Delta t}}{\partial x_i^t} \frac{\partial u_k^{t+\Delta t}}{\partial x_j^t} \right).$$

Decompose the unknown stresses and strains as:

$$S_{ij}^{t+\Delta t} = S_{ij}^t + S_{ij} = \sigma_{ij}^t + S_{ij},$$

$$E_{ij}^{t+\Delta t} = E_{ij}^t + E_{ij} = E_{ij}.$$

This is because:

$$S_{ij}^t = J^t F^{-1(t)}(x^t) \sigma_{ij}^t(x^t) F^t(x^t) = \sigma_{ij}^t(x^t),$$

and

$$E_{ij}^t = \frac{1}{2} \left(\frac{\partial u_j^t}{\partial x_i^t} + \frac{\partial u_i^t}{\partial x_j^t} + \frac{\partial u_k^t}{\partial x_i^t} \frac{\partial u_k^t}{\partial x_j^t} \right).$$

Since now u_j^t denotes the displacement of the particle in the body from time t (reference time) to t . Thus

$$u_j^t = 0,$$

$$E_{ij}^t = 0,$$

$$E_{ij} = \frac{1}{2} \left(\frac{\partial u_j}{\partial x_i^t} + \frac{\partial u_i}{\partial x_j^t} + \frac{\partial u_k^t}{\partial x_i^t} \frac{\partial u_k}{\partial x_j^t} + \frac{\partial u_k}{\partial x_i^t} \frac{\partial u_k^t}{\partial x_j^t} \right) + \frac{1}{2} \left(\frac{\partial u_k}{\partial x_i^t} \frac{\partial u_k}{\partial x_j^t} \right) = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j^t} + \frac{\partial u_j}{\partial x_i^t} \right) + \frac{1}{2} \frac{\partial u_k}{\partial x_j^t} \frac{\partial u_k}{\partial x_i^t}.$$

Therefore,

$$E_{ij}^{t+\Delta t} = E_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j^t} + \frac{\partial u_j}{\partial x_i^t} \right) + \frac{1}{2} \frac{\partial u_k}{\partial x_j^t} \frac{\partial u_k}{\partial x_i^t}.$$

Thus there is no initial displacement effect in this expression. We define the linear strain increment:

$$\theta_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j^t} + \frac{\partial u_j}{\partial x_i^t} \right),$$

and the nonlinear strain increment

$$\eta_{ij} = \frac{1}{2} \frac{\partial u_k}{\partial x_j^t} \frac{\partial u_k}{\partial x_i^t}.$$

Hence,

$$E_{ij} = \theta_{ij} + \eta_{ij},$$

$$\delta E_{ij} = \delta \theta_{ij} + \delta \eta_{ij}.$$

The equation of the principle of virtual work becomes

$$\int_{\Omega^t} S_{ij} \delta E_{ij} dx + \int_{\Omega^t} \sigma_{ij}^t \delta \eta_{ij} dx = \mathcal{R}^{t+\Delta t} - \int_{\Omega^t} \sigma_{ij}^t \delta \theta_{ij} dx.$$

Given a variation, the right hand side is known. So far, no approximation has been made. Just like in the T.L. case, we begin to linearize (with respect to unknown incremental displacement) terms.

The term $\int_{\Omega^t} \sigma_{ij}^t \delta \eta_{ij} dx$ is linear in u , the reason is the same as in the T.L. case. The term $\int_{\Omega^t} S_{ij} \delta E_{ij} dx$ contains a nonlinear function. We repeat the linearization procedure from the T.L. case, to get:

$$S_{ij} \cong \frac{\partial S_{ij}^t}{\partial E_{rs}^t} \theta_{rs},$$

and

$$S_{ij} \delta E_{ij}^t \cong \frac{\partial S_{ij}^t}{\partial E_{rs}^t} \theta_{rs} \delta \theta_{ij}.$$

Till now, we have derived the final linearized equation

$$\int_{\Omega^t} \frac{\partial S_{ij}^t}{\partial E_{rs}^t} \theta_{rs} \delta \theta_{ij} dx + \int_{\Omega^t} \sigma_{ij}^t \delta \eta_{ij} dx = \mathcal{R}^{t+\Delta t} - \int_{\Omega^t} \sigma_{ij}^t \delta \theta_{ij} dx.$$

We find that in the U.L. formulation, we work with the actual physical stress (Cauchy stress) .

Lagrangian elasticity tensor

In the previous section we have presented two linearization formulations which are both suitable for large displacement, large rotation and large strain problem analysis. To use these formulations, appropriate constitutive relations must be employed. In this section, we would like to derive a specific form of the elasticity tensors $\mathcal{C}_{ijrs} \left(= \frac{\partial S_{ij}^t}{\partial E_{rs}^t} \right)$ appearing in both formulations.

First, we would like to point out that the elastic constitutive relations which are suitable for small displacements, small rotation and small strain analysis (like Hooke's law) can be directly used in large displacement, large rotation but small strain analysis. Specifically, we have that constitutive relation holds for infinitesimal analysis,

$$\sigma_{ij} = \mathcal{C}_{ijrs} \epsilon_{rs},$$

where ϵ_{rs} is Cauchy's infinitesimal strain tensor and σ_{ij} is Cauchy stress tensor, and \mathcal{C}_{ijrs} is a constant forth-order elasticity tensor. For example,

$$\mathcal{C}_{ijrs} = \lambda \delta_{ij} \delta_{rs} + \mu (\delta_{ir} \delta_{js} + \delta_{is} \delta_{jr})$$

is the elasticity tensor for Hookean solid where λ and μ are Lamé constants. Then for large displacement, large rotation but small strain analysis,

$$S_{ij} = \mathcal{C}_{ijrs} E_{rs},$$

where S_{ij} is the second Piola-Kirchhoff stress tensor and E is Green's strain tensor. The reason is explained as follows: separate the large displacement, large rotation but small strain deformation into two steps. Firstly, the body experiences small displacement, small rotation and small strain deformation. Thus at this stage,

$$S = J F^{-1} \sigma F^{-T} \cong \sigma, \quad E \cong \epsilon, \quad S_{ij} = \mathcal{C}_{ijrs} E_{rs}.$$

Next the body experiences rigid body motions (rotation and translations). Since the S and E stay invariant under rigid motions (which has been mentioned in section 3.2.1), the relation still holds for large displacement, large rotation but small strain analysis. Now, we also recognize that the Hookean solid constitutive relation in large displacement and large rotation but small strain analysis is the same as in the St.Venant-Kirchhoff model (from section 3.2.2).

Next, we discuss the form of the elasticity tensor in the updated Lagrangian formulation. Note that we would like to have the updated Lagrangian formulation to produce the same set of linear equations as in the total Lagrangian formulation (presuming that we use the same discretization procedure). Recall the definition of Green's strain tensor:

$$ds^2 - ds_0^2 = 2E_{ij} dX_i dX_j,$$

where ds and ds_0 represent the length of an infinitesimal line element in the body, in the current and original configuration respectively. Thus, we have:

$$\begin{aligned} E_{ij}^t dX_i dX_j &= \frac{1}{2} (ds^2 - ds_0^2), \\ E_{ij}^{t+\Delta t} dX_i dX_j &= \frac{1}{2} (d(s')^2 - ds_0^2), \end{aligned}$$

where s' represents the length of an infinitesimal line element in the body in the $t + \Delta t$ configuration. By deducting the first equation from the second equation we have:

$$E_{ij}^0 dX_i dX_j = \frac{1}{2} (d(s')^2 - ds^2),$$

where the 0 at the superscript place indicates that the variable is evaluated with respect to the original configuration. At the same time if we put the configuration at time t as the reference configuration, we have

$$E_{ij} dx_i dx_j = \frac{1}{2} (d(s')^2 - ds^2),$$

where E_{ij} is the incremental strain with respect to the current time t configuration. Hence,

$$E_{ij}^0 dX_i dX_j = E_{rs} dx_r dx_s = \frac{1}{2} (d(s')^2 - ds^2).$$

Using $d\vec{x} = F d\vec{X}$, we obtain that

$$E_{ij}^0 dX_i dX_j = E_{rs} F_{ri} F_{sj} dX_i dX_j.$$

Since this relation holds for any infinitesimal line element, we have

$$E_{ij}^0 = E_{rs} F_{ri} F_{sj}.$$

Note that

$$dS_{ij}^0 = C_{ijrs}^0 dE_{rs}^0,$$

where

$$C_{ijrs}^0 = \frac{\partial S_{ij}^0}{\partial E_{rs}^0}$$

is the Lagrangian elasticity tensor with respect to the original configuration at time t . Take the relation with respect to the configuration at time t . Then we have:

$$dS_{ij} = C_{ijrs} dE_{rs}.$$

Note that

$$dS_{ij}^0 = J F_{im}^{-1} F_{jn}^{-1} dS_{mn},$$

$$dE_{rs}^0 = F_{pr} F_{qs} dE_{pq}.$$

Hence,

$$dS_{ij}^0 = JF_{im}^{-1}F_{jn}^{-1}dS_{mn} = C_{ijrs}^0 dE_{rs}^0 = C_{ijrs}^0 (F_{pr}F_{qs}dE_{pq}).$$

Thus we have,

$$dS_{ab} = JF_{ai}F_{bj}C_{ijrs}^0 F_{pr}F_{qs}dE_{pq} = C_{abpq}dE_{pq}.$$

We have got the form for the elasticity tensor in updated Lagrangian formulation as:

$$C_{abpq} = JF_{ai}F_{bj}C_{ijrs}^0 F_{pr}F_{qs}dE_{pq}.$$

Theoretically and numerically, if formulated mathematically correct, the total Lagrangian and updated Lagrangian formulation yield exactly the same result. However, integration of constitutive equations for certain types of material behaviour makes the implementation of the total Lagrangian formulation inconvenient. If the constitutive equations are converted back to the original configuration and proper transformations are applied, then both formulations are equivalent [35]. Therefore, the choice of formulations (T.L. or U.L.) is based merely on the numerical effectiveness of the methods. But if the stress-strain law is available in terms of the S (the second Piola-Kirchhoff stress tensor), the T.L. formulation will be most effective in general. Note that it is possible that more than one formulation is used in one continuum. The user does not have absolute freedom in choosing the formulations in Marc. Marc chooses the formulations dependent on the formulation of the element in some cases.

Constitutive equations

Three types of material models have been tried out in this project. They are the St.Venant-Kirchhoff model, Mooney model and Gent model. Note that they are all hyperelastic models (instead of linear model), the reason for doing so is that: though we know that the linear model gives formulation and computation simplicity, the physical behaviour of soft tissue will only be considered as linear elastic if its displacement and deformation remain small (typically less than 10%) of the mesh size [17]. Presume the mesh size is $3mm$, then it means that the linear elastic model is only suitable if the displacement is less than $0.3mm$. This is not true for the nodes locating at the lower face, where they usually have a $5\sim 8mm$ displacement during the mouth opening process. Below I will give the constitutive equations of the three materials in turn.

- St.Venant-Kirchhoff model

As we have mentioned in the previous section, the St.Venant-Kirchhoff model is an extension of Hookean solid to large displacement and large rotation regime. Its constitutive relation is given as:

$$S = \lambda(trE)I + 2\mu E,$$

or alternatively,

$$S_{ij} = C_{ijrs}E_{rs},$$

with

$$C_{ijrs} = \lambda\delta_{ij}\delta_{rs} + \mu(\delta_{ir}\delta_{js} + \delta_{is}\delta_{jr}),$$

where λ and μ are Lamé constants. Obviously, the Lagrangian elasticity tensor in this case is a constant tensor, C_{ijrs} .

- Mooney-Rivlin model

This material model is often being used for modelling rubber, and assumes near incompressibility. As mentioned in section 3.2.2, for incompressible Mooney material the

strain density functions can be rewritten as:

$$\Psi(C) = C_1(\widehat{I}_C - 3) + C_2(\widehat{II}_C - 3) + U(III_C),$$

where $\widehat{I}_C = III_C^{-1/3} I_C$, and $\widehat{II}_C = III_C^{-1/3} II_C$ and in Marc the volumetric part of the strain density function U is given by

$$U(III_C) = \frac{9}{2} k \left(III_C^{\frac{1}{3}} - 1 \right)^2,$$

where k represents the bulk modulus and is calculated by

$$k = 1000 * (C_{01} + C_{10}).$$

- Gent model

Another material model that we used to simulate the skin layer is the Gent model. Just like the Mooney model, the material is assumed to be nearly incompressible. Its strain density function is:

$$\Psi(C) = -\frac{E}{2} I_m \ln \left(1 - \frac{\widehat{I}_C - 3}{I_m} \right) + U(III_C),$$

where E is the small strain tensile modulus, and I_m is the maximum value of the first variant and thus is demanded to be larger than 3.

The Lagrangian or material elasticity tensor can be obtained from the strain energy density function. Intuitively speaking, this fourth-order tensor gives the tangent value of the nonlinear “strain-stress curve”. It is defined as :

$$\mathcal{C} = \frac{\partial S}{\partial E} = 2 \frac{\partial S}{\partial C} = 4 \frac{\partial^2 \Psi}{\partial C \partial C}.$$

To write out explicitly,

$$\mathcal{C} = \sum_{i,j,k,l=1}^3 C_{ijkl} \varepsilon_i \otimes \varepsilon_j \otimes \varepsilon_k \otimes \varepsilon_l,$$

where $\varepsilon_k, \varepsilon_l, \varepsilon_i \varepsilon_j$ are the Cartesian basis vectors, “ \otimes ” denotes tensor product and

$$C_{ijkl} = \frac{\partial S_{ij}}{\partial E_{kl}} = 4 \frac{\partial^2 \Psi}{\partial C_{ij} \partial C_{kl}} = C_{klij}.$$

Note that combining the derivation of the elasticity tensor used in the updated Lagrangian formulation from the last section, we are also able to derive the elasticity tensor used in the updated Lagrangian formulation.

The reasons that we choose the three particular hyperelastic material models to model skin is the following: the St.Venant-Kirchhoff model can be seen as the most “linear” hyperelastic model, although it is only suitable for the small strain analysis, it still holds a fair ground to model the skin layer for mouth-opening process, considering the not so big stretching in most of the skin during the process. As for Mooney material, this is a very common hyperelastic material model for modelling rubber and it also used in several literature to simulate biological tissues. Gent model’s uniaxial result looks quite promising (quite similar to the one’s from human skin), thus we expect that the Gent model can show its advantage as the strain becomes bigger in the simulation. Another important reason for choosing Gent and Mooney material models are their availability in the material library of Marc. For other hyperelastic material models (like Yeoh), the

user needs to write a subroutine to implement it.

4.1.4. Finite element approach

In this section, we will present a detailed picture on how the finite element method has been applied to solve the linearized problem posed above. To start with, we show a general procedure on how to solve a nonlinear problem using FEM.

General FEM procedure

In the finite element method, the actual continuum (in our case, the skin layer) is represented as an assemblage of subdivisions called finite elements. These elements are considered to be interconnected at specified joints called nodes. Since the actual value of the field variable (like displacement) inside the continuum is unknown, we assume that the value of the field variable inside a finite element can be approximated by simple functions. These approximating functions are defined based on the nodes positions. When the equilibrium equations for the continuum are written, the new unknowns will be the nodal values of the field variable. By solving the equilibrium equations, which are generally in the form of matrix equations for linearized nonlinear problems, the nodal values of the field variable will be obtained. Once these are known, the approximation of the field variable throughout the continuum is known.

A problem is nonlinear if the force-displacement relationship depends on the current state (that is, current displacement, force, and stress-strain relations). Let u be a generalized displacement vector, P a generalized force vector, and K the stiffness matrix. The expression of the force-displacement relation for a nonlinear problem is

$$K(u, P)u = P.$$

There are three sources of nonlinearity: material, geometrical, and boundary conditions.

Material (physical) nonlinearity results from the nonlinear relationship between stresses and strains. For example, the Mooney model and Gent model have non-linear stress-strain relationships since their material elasticity tensor is not constant.

Geometrical nonlinearity arises from the nonlinear relation between strain and displacement. Since we choose to use the Lagrangian formulation and therefore the Green strain tensor (instead of Cauchy's infinitesimal strain tensor) appears in the principle of virtual work formulation. This way, the geometrical nonlinearity is incorporated.

Contact problem, friction and nonlinear support can cause nonlinearity via the boundary conditions. We use linear springs as our support condition, thus we do not introduce nonlinearity from boundary conditions.

The solution of a general nonlinear problem using the finite element method follows a step-by-step process, stated as follows:

1. **Discretization of the continuum**

The first step in the finite element method is to divide the continuum body into elements. Hence, the body of interest is to be modelled with suitable finite elements. The number, type, and arrangement of the elements are to be chosen.

2. **Selection of a proper interpolation function**

We approximate the solution with the linear combination of interpolation functions within an element. The interpolation functions must be simple from a computational standpoint, but it should satisfy certain convergence requirements. In general, the interpolation function is taken in the form of a polynomial.

3. Derivation of element tangent stiffness matrices and out of balance nodal force vectors

From the assumed interpolation function, we are ready to derive the tangent stiffness matrix K^t and the out of balance nodal force vector f^t of element e from the total/updated Lagrangian formulation (linearized continuum mechanics equation) respectively.

4. Assemblage of element equations to obtain the overall equilibrium equations

The individual element tangent stiffness matrices and load vectors are to be assembled in a suitable manner in this step. The overall equilibrium equation has the formulation:

$$K^t \Delta U = f^t,$$

where K^t is the assembled tangent stiffness matrix, ΔU the incremental vector of nodal displacements, and f^t the assembled out of balance nodal forces vector of the whole continuum.

5. Solution for the unknown nodal displacements

To obtain the solution of a nonlinear problem we use Newton-Raphson iterative solution. Namely,

$$K^t \Delta U = f^t = R^{t+\Delta t} - F^t,$$

$$U^{t+\Delta t} = U + \Delta U.$$

Note that within one time step, one usually performs multiple times increment solutions to achieve a certain accuracy. $R^{t+\Delta t}$ is the externally applied load vector at time $t + \Delta t$. F^t is the vector of nodal point forces related with the internal stresses at time t .

Following the steps proposed above, we are ready to present the finite element approach tailored to our problems.

Discretization of the continuum

As it has been shown in the configuration of the models, we discretize the skin layer uniformly. There are three types of elements we use in this project:

- Four-node thick-shell element and eight-node element
- Isoparametric, arbitrary hexahedral element
- Three-dimensional arbitrarily distorted brick, Herrmann formulation

Illustrations are given in Figure 4.4, Figure 4.5 and Figure 4.6.

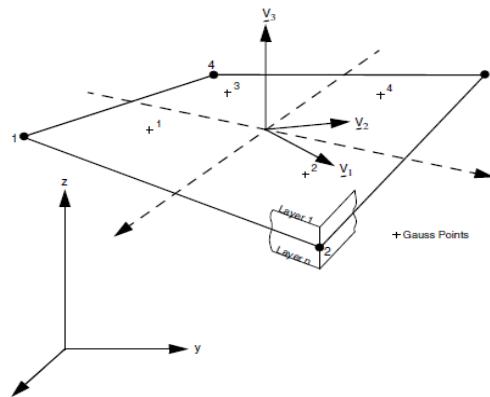


Figure 4.4 Shell element

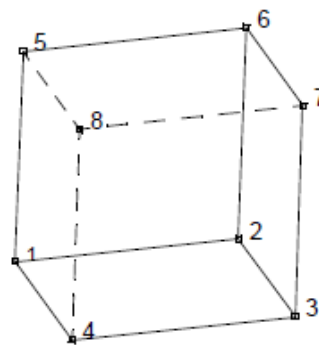


Figure 4.5 Hexahedral element

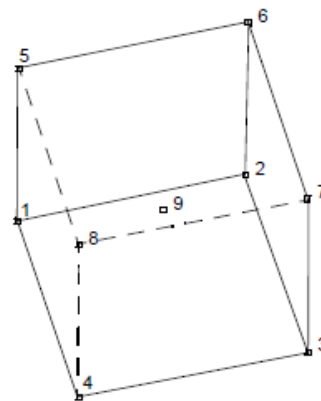


Figure 4.6 Hexahedral element, Herrmann formulation

Selection of a proper interpolation function

In this section, we will present the three types of elements with more details. The isoparametric, arbitrary hexahedral element is our focus and will be given in the full details.

Shell element

This is a structural element and has global displacements and rotations as degrees of freedom. Bilinear interpolation is used for the coordinates, displacements and the rotations. Nodal thickness can be defined to create certain “virtual” thickness. Based on the consideration that it

has quite distinctive properties compared with the continuum element and it does not produce a better result than the continuum 3D isoparametric hexahedral element, the study about this element is very limited in this work and the discussion about it will be ended here. More information about this type of element can be found in [37,38]. Note that the skull layer shown in section 4.1.2, is discretized using shell elements and is assumed to be rigid and bearing the thickness $0.2mm$.

Three-dimensional arbitrarily distorted brick, Herrmann Formulation

In the section **Incompressible material and near incompressibility**, we have mentioned that a special formulation is needed to handle the incompressibility or near incompressibility within finite element context. Herrmann's formulation is a solution for it. Below we will give an introduction to Herrmann formulation for linear problems. The treatment of nonlinear problems (which is the case in all our models) will not be dealt with here. Further information on this topic can be found in [38].

We know that the strain is a measure of deformation, thus we expect the strain can also reflect the volume change of the body during the deformation. The volumetric strain, ϵ_V , is a strain measure to measure the relative dilation of the body. In the infinitesimal strain theory, it can be shown that

$$\epsilon_V = \frac{\Delta V}{V} = \epsilon_{11} + \epsilon_{22} + \epsilon_{33},$$

where ϵ_{kk} ($k = 1,2,3$.) are the components of Cauchy's infinitesimal tensor ϵ_{ij} . Thus the incompressibility property actually poses a kinematic constraint to the equilibrium equation as:

$$\text{div } u = \epsilon_V = 0.$$

The constitutive relation for Hookean solid can be written as:

$$\sigma_{ij} = \lambda \epsilon_V \delta_{ij} + 2\mu \epsilon_{ij}.$$

with

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

We see that, when the material approaches incompressibility, $\epsilon_V \rightarrow 0$, and $\lambda \rightarrow \infty$ (since $\nu \rightarrow 0.5$). In Herrmann formulation, a new (independent) variable p is introduced. It is defined as:

$$p = -\lambda \epsilon_V.$$

Thus the new constitutive relation can be written as:

$$\sigma_{ij} = -p \delta_{ij} + 2\mu \epsilon_{ij}.$$

and since we introduce a new variable p , we also need another equation for it as:

$$\text{div } u + \frac{p}{\lambda} = 0.$$

Note that in the incompressible case, we can interpret p as the hydrostatic pressure, since the hydrostatic pressure is defined as $-\sigma_{ii}/3$. While in the nearly incompressible case,

$$-\frac{\sigma_{ii}}{3} = -\left(\lambda + \frac{2\mu}{3}\right) \text{div } u.$$

Notice that

$$p = -\lambda \operatorname{div} u,$$

we see that when $\mu \ll \lambda$ (nearly incompressible), p is still a good approximation to the hydrostatic pressure and $p = -\sigma_{ii}/3$, for incompressible material.

Numerical experiments show that for the nearly incompressible material (which means the value of λ is large), a not fine enough mesh will lead to an inaccurate prediction of volumetric strain (which is approaching to 0 in this case), and consequently, a large error in the stress calculation. In this case the finite elements are *locking* [38]. The invention of Herrmann formulation avoids the locking of elements.

Coming back to the element itself, this is an eight-node, isoparametric element with an additional ninth node for the pressure. The element is based on a trilinear interpolation function in $\tilde{\xi}$ space. The pressure is assumed constant throughout the element. The 24 generalized displacements are related to the x, y, z -displacements (in global coordinates) at the eight corners of the distorted cube. The last node has one degree of freedom (negative hydrostatic pressure).

Note that if we change the formulation of the problem (by adding another variable p), all the related aspects need to be changed as well (e.g. the formulation of the principle of virtual work and accordingly the stiffness matrix and out of balance nodal force vector, etc.). These changes are not the focus of the thesis. The discussion of the Herrmann formulation will be ended here.

Isoparametric, arbitrary hexahedral element

This type of elements uses trilinear interpolation functions. The isoparametric, arbitrary hexahedral element is the basic element for three-dimensional analysis. The domain Ω^e in x, y, z -space is the image of the tri-unit cube in ξ -space under the trilinear mapping

$$x(\tilde{\xi}) = \alpha_0 + \alpha_1\xi + \alpha_2\eta + \alpha_3\zeta + \alpha_4\xi\eta + \alpha_5\eta\zeta + \alpha_6\xi\zeta + \alpha_7\xi\zeta\eta,$$

where $\tilde{\xi} = (\xi, \eta, \zeta)$. Similar expressions hold for $y(\tilde{\xi})$ and $z(\tilde{\xi})$. The coefficients $\alpha_0, \dots, \alpha_7$ are determined via the conditions:

$$x(\tilde{\xi}_a) = x_a^e, \quad a = 1, \dots, 8,$$

where $\tilde{\xi}_a$ are the nodal points coordinates in ξ -space, and x_a^e are the corresponding coordinates of the nodal points in x, y, z -space (x direction). This gives rise to a system of linear algebraic equations. Solving for the α 's and substituting in the mapping yields

$$x(\tilde{\xi}) = \sum_{a=1}^8 N_a(\tilde{\xi}) x_a^e,$$

where

$$N_a(\xi, \eta, \zeta) = \frac{1}{8} (1 + \xi_a \xi) (1 + \eta_a \eta) (1 + \zeta_a \zeta),$$

with similar expressions for $y(\tilde{\xi})$ and $z(\tilde{\xi})$. The correspondence between (ξ_a, η_a, ζ_a) and nodal points in ξ -space are given in the following table.

a	ξ_a	η_a	ζ_a
1	-1	-1	-1
2	1	-1	-1
3	1	1	-1
4	-1	1	-1
5	-1	-1	1

6	1	-1	1
7	1	1	1
8	-1	1	1

Table 1 Node numbers and coordinates correspondence in natural coordinate system

Once the element type has been decided, by invoking the isoparametric concept, the discretization of the displacement field can also be decided. The displacement field within each element is interpolated as:

$$u^h(\vec{\xi}) = \begin{bmatrix} u_1^h \\ u_2^h \\ u_3^h \end{bmatrix} = \begin{bmatrix} \sum_{a=1}^8 N_a(\vec{\xi}) d_{a1}^{(e)} \\ \sum_{a=1}^8 N_a(\vec{\xi}) d_{a2}^{(e)} \\ \sum_{a=1}^8 N_a(\vec{\xi}) d_{a3}^{(e)} \end{bmatrix} = \sum_{a=1}^8 N_a \mathbb{d}_a^{(e)} = \mathbb{N} \mathbb{d}^{(e)},$$

where

$$\mathbb{N} = [\mathbb{N}_1, \mathbb{N}_2, \mathbb{N}_3, \mathbb{N}_4, \mathbb{N}_5, \mathbb{N}_6, \mathbb{N}_7, \mathbb{N}_8], \mathbb{N}_a = \begin{bmatrix} N_a & 0 & 0 \\ 0 & N_a & 0 \\ 0 & 0 & N_a \end{bmatrix}, \mathbb{d}^{(e)} = \begin{bmatrix} \mathbb{d}_1^{(e)} \\ \mathbb{d}_2^{(e)} \\ \mathbb{d}_3^{(e)} \\ \mathbb{d}_4^{(e)} \\ \mathbb{d}_5^{(e)} \\ \mathbb{d}_6^{(e)} \\ \mathbb{d}_7^{(e)} \\ \mathbb{d}_8^{(e)} \end{bmatrix}, \mathbb{d}_a^{(e)} = \begin{bmatrix} d_{a1}^{(e)} \\ d_{a2}^{(e)} \\ d_{a3}^{(e)} \end{bmatrix},$$

$\mathbb{d}_a^{(e)}$ is the displacement vector at node a in the element e . Note that the same interpolation is used for the three spatial components of the displacement vector.

Besides displacement field discretization, another frequently used matrix is (Cauchy's infinitesimal) strain-displacement relation matrix \mathbb{B} . The derivation of \mathbb{B} is presented below. The (Cauchy's infinitesimal) strain field is linked to the displacement by the relation $\epsilon = \frac{1}{2}(\nabla u + \nabla u^T)$. Making use of the symmetry of strain tensor, an alternative way of writing it is:

$$\begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ 2\epsilon_{12} \\ 2\epsilon_{23} \\ 2\epsilon_{31} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix},$$

where u_1, u_2, u_3 are the three spatial components of the displacement vector u . Combining with the displacement discretization result $u^h(\vec{\xi}) = \mathbb{N} \mathbb{d}^{(e)}$, we get that the strain expression throughout

one element is:

$$\epsilon^h = \begin{bmatrix} \epsilon_{11}^h \\ \epsilon_{22}^h \\ \epsilon_{33}^h \\ 2\epsilon_{12}^h \\ 2\epsilon_{23}^h \\ 2\epsilon_{13}^h \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} u_1^h \\ u_2^h \\ u_3^h \end{bmatrix} = \sum_{a=1}^8 \begin{bmatrix} \frac{\partial N_a}{\partial x} d_{a1}^{(e)} \\ \frac{\partial N_a}{\partial y} d_{a2}^{(e)} \\ \frac{\partial N_a}{\partial z} d_{a3}^{(e)} \\ \frac{\partial N_a}{\partial y} d_{a1}^{(e)} + \frac{\partial N_a}{\partial x} d_{a2}^{(e)} \\ \frac{\partial N_a}{\partial z} d_{a2}^{(e)} + \frac{\partial N_a}{\partial y} d_{a3}^{(e)} \\ \frac{\partial N_a}{\partial z} d_{a1}^{(e)} + \frac{\partial N_a}{\partial x} d_{a3}^{(e)} \end{bmatrix} = \sum_{a=1}^8 \mathbb{B}_a d_a^{(e)} = \mathbb{B} d^{(e)},$$

where \mathbb{B} is the element strain matrix given by

$$\mathbb{B} = [\mathbb{B}_1, \mathbb{B}_2, \mathbb{B}_3, \mathbb{B}_4, \mathbb{B}_5, \mathbb{B}_6, \mathbb{B}_7, \mathbb{B}_8].$$

\mathbb{B}_a is the strain matrix of node a , with

$$\mathbb{B}_a = \begin{bmatrix} \frac{\partial N_a}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_a}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_a}{\partial z} \\ \frac{\partial N_a}{\partial y} & \frac{\partial N_a}{\partial x} & 0 \\ 0 & \frac{\partial N_a}{\partial z} & \frac{\partial N_a}{\partial y} \\ \frac{\partial N_a}{\partial z} & 0 & \frac{\partial N_a}{\partial x} \end{bmatrix}, \quad a = 1, \dots, 8.$$

Next, let us compute \mathbb{B}_a for each node, we note that N_a are given in natural coordinate system $\bar{\xi}$. To compute each element in \mathbb{B}_a , recall the definition of an isoparametric element:

$$x(\bar{\xi}) = \sum_{a=1}^8 N_a(\bar{\xi}) x_a^e, \quad y(\bar{\xi}) = \sum_{a=1}^8 N_a(\bar{\xi}) y_a^e, \quad z(\bar{\xi}) = \sum_{a=1}^8 N_a(\bar{\xi}) z_a^e.$$

It allows us to obtain a relationship between the derivatives of the shape functions with respect to the Cartesian coordinates and with respect to the natural coordinates. Note that N_a is expressed in terms of natural coordinates $\bar{\xi}$. The chain rule of derivatives yields:

$$\begin{aligned} \frac{\partial N_a}{\partial \xi} &= \frac{\partial N_a}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial N_a}{\partial y} \frac{\partial y}{\partial \xi} + \frac{\partial N_a}{\partial z} \frac{\partial z}{\partial \xi}, \\ \frac{\partial N_a}{\partial \eta} &= \frac{\partial N_a}{\partial x} \frac{\partial x}{\partial \eta} + \frac{\partial N_a}{\partial y} \frac{\partial y}{\partial \eta} + \frac{\partial N_a}{\partial z} \frac{\partial z}{\partial \eta}, \\ \frac{\partial N_a}{\partial \zeta} &= \frac{\partial N_a}{\partial x} \frac{\partial x}{\partial \zeta} + \frac{\partial N_a}{\partial y} \frac{\partial y}{\partial \zeta} + \frac{\partial N_a}{\partial z} \frac{\partial z}{\partial \zeta}, \end{aligned}$$

which in matrix form reads:

$$\begin{bmatrix} \frac{\partial N_a}{\partial \xi} \\ \frac{\partial N_a}{\partial \eta} \\ \frac{\partial N_a}{\partial \zeta} \end{bmatrix} = J^{(e)} \begin{bmatrix} \frac{\partial N_a}{\partial x} \\ \frac{\partial N_a}{\partial y} \\ \frac{\partial N_a}{\partial z} \end{bmatrix},$$

where

$$J^{(e)} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix}$$

is the Jacobian transformation matrix of the derivatives of N_a between the natural and global coordinate systems. The superscript (e) indicates that the matrix is always computed at element level. We deduced that

$$\begin{bmatrix} \frac{\partial N_a}{\partial x} \\ \frac{\partial N_a}{\partial y} \\ \frac{\partial N_a}{\partial z} \end{bmatrix} = [J^{(e)}]^{-1} \begin{bmatrix} \frac{\partial N_a}{\partial \xi} \\ \frac{\partial N_a}{\partial \eta} \\ \frac{\partial N_a}{\partial \zeta} \end{bmatrix}.$$

The terms of $J^{(e)}$ can be computed by the isoparametric definition, i.e.

$$x(\tilde{\xi}) = \sum_{a=1}^8 N_a(\tilde{\xi}) x_a^e,$$

$$\frac{\partial x}{\partial \xi} = \sum_{a=1}^8 \frac{\partial N_a(\tilde{\xi})}{\partial \xi} x_a^e, \quad \frac{\partial x}{\partial \eta} = \sum_{a=1}^8 \frac{\partial N_a(\tilde{\xi})}{\partial \eta} x_a^e, \quad \frac{\partial x}{\partial \zeta} = \sum_{a=1}^8 \frac{\partial N_a(\tilde{\xi})}{\partial \zeta} x_a^e, \text{ etc.}$$

Hence,

$$J^{(e)}(\tilde{\xi}) = \sum_{a=1}^8 \begin{bmatrix} \frac{\partial N_a(\tilde{\xi})}{\partial \xi} x_a^e & \frac{\partial N_a(\tilde{\xi})}{\partial \xi} y_a^e & \frac{\partial N_a(\tilde{\xi})}{\partial \xi} z_a^e \\ \frac{\partial N_a(\tilde{\xi})}{\partial \eta} x_a^e & \frac{\partial N_a(\tilde{\xi})}{\partial \eta} y_a^e & \frac{\partial N_a(\tilde{\xi})}{\partial \eta} z_a^e \\ \frac{\partial N_a(\tilde{\xi})}{\partial \zeta} x_a^e & \frac{\partial N_a(\tilde{\xi})}{\partial \zeta} y_a^e & \frac{\partial N_a(\tilde{\xi})}{\partial \zeta} z_a^e \end{bmatrix}.$$

The computations above enable us to compute strain matrix for each isoparametric hexahedral element.

Derivation of element tangent stiffness matrices and out of balance nodal force vectors

Since we are dealing with nonlinear problems, element tangent stiffness matrix instead of

element stiffness matrix will be obtained as needed for the iteration procedure. Below we start with the continuous linearized equation (T.L. & U.L.). Eventually, we will get the discretized counterparts (matrices or vectors) for every term that appears in the continuous equation.

Discretized total Lagrangian formulation

In section 4.1.3, we have derived the total Lagrangian formulation which is suitable for large displacement, large rotation and large strain analysis. It reads

$$\int_{\Omega^0} \frac{\partial S_{ij}^t}{\partial E_{rs}^t} \theta_{rs} \delta \theta_{ij} dX + \int_{\Omega^0} S_{ij}^t \delta \eta_{ij} dX = \mathcal{R}^{t+\Delta t} - \int_{\Omega^0} S_{ij}^t \delta \theta_{ij}^t dX.$$

We divide the domain into elements, on each element, it holds:

$$\int_{\Omega^e} \frac{\partial S_{ij}^t}{\partial E_{rs}^t} \theta_{rs} \delta \theta_{ij} dX + \int_{\Omega^e} S_{ij}^t \delta \eta_{ij} dX = \mathcal{R}_e^{t+\Delta t} - \int_{\Omega^e} S_{ij}^t \delta \theta_{ij}^t dX.$$

where Ω^e denotes the element domain and $\mathcal{R}_e^{t+\Delta t}$ denotes the external work applied on the element e .

Now we are going to discretize every term in this formulation. In the last section, we have discretised the displacement field and derived the (Cauchy infinitesimal) strain-displacement relation matrix within an element e . They read:

$$u^h(\tilde{\xi}) = \mathbb{N}\mathbb{d}^{(e)},$$

$$\epsilon^h = \mathbb{B}\mathbb{d}^{(e)}.$$

Be aware that we are using the total Lagrangian formulation, thus all the variables are referred to the original configuration \tilde{X} , thus the matrix \mathbb{B} should have the form:

$$\mathbb{B} = [\mathbb{B}_1, \mathbb{B}_2, \mathbb{B}_3, \mathbb{B}_4, \mathbb{B}_5, \mathbb{B}_6, \mathbb{B}_7, \mathbb{B}_8],$$

with

$$\mathbb{B}_a = \begin{bmatrix} \frac{\partial N_a}{\partial X} & 0 & 0 \\ 0 & \frac{\partial N_a}{\partial Y} & 0 \\ 0 & 0 & \frac{\partial N_a}{\partial Z} \\ \frac{\partial N_a}{\partial Y} & \frac{\partial N_a}{\partial X} & 0 \\ 0 & \frac{\partial N_a}{\partial Z} & \frac{\partial N_a}{\partial Y} \\ \frac{\partial N_a}{\partial Z} & 0 & \frac{\partial N_a}{\partial X} \end{bmatrix}, \quad a = 1, \dots, 8.$$

The components of \mathbb{B}_a are the partial derivatives of N_a with respect to the original configuration \tilde{X} . There is no technical difficulty in taking partial derivatives with respect to the original configuration, since the original configuration is known. Using this displacement discretization, the first term at the left hand side of the continuous equation $\int_{\Omega^e} \frac{\partial S_{ij}^t}{\partial E_{rs}^t} \theta_{ij} \delta \theta_{ij} dX$, is approximated as:

$$\int_{\Omega^e} \frac{\partial S_{ij}^t}{\partial E_{rs}^t} \theta_{ij} \delta \theta_{ij} dX \rightarrow \delta \mathbb{d}^{(e)} \left(\int_{\Omega^0} (\mathbb{B}_L^t)^T \mathbb{C} \mathbb{B}_L^t dX \right) \mathbb{d}^{(e)},$$

where \mathbb{C} is the matrix converting form for the fourth order Lagrangian material elasticity tensor ($\mathbb{C}_{ijrs} = \frac{\partial S_{ij}^t}{\partial E_{rs}^t}$). $\delta \mathbb{d}^{(e)}$ denotes the virtual nodal displacement as

$$\delta \mathbb{d}^{(e)} = \begin{bmatrix} \delta \mathbb{d}_1^{(e)} \\ \delta \mathbb{d}_2^{(e)} \\ \delta \mathbb{d}_3^{(e)} \\ \delta \mathbb{d}_4^{(e)} \\ \delta \mathbb{d}_5^{(e)} \\ \delta \mathbb{d}_6^{(e)} \\ \delta \mathbb{d}_7^{(e)} \\ \delta \mathbb{d}_8^{(e)} \end{bmatrix}, \quad \delta \mathbb{d}_a^{(e)} = \begin{bmatrix} \delta d_{a1}^{(e)} \\ \delta d_{a2}^{(e)} \\ \delta d_{a3}^{(e)} \end{bmatrix}.$$

Note that the subscript L denotes that this is the matrix involving linear strain increment θ_{ij} . \mathbb{B}_L^t represents the linear strain part θ_{ij} -displacement relation matrix. Namely, we want to achieve:

$$\theta^h = \begin{bmatrix} \theta_{11}^h \\ \theta_{22}^h \\ \theta_{33}^h \\ 2\theta_{12}^h \\ 2\theta_{23}^h \\ 2\theta_{13}^h \end{bmatrix} = \sum_{a=1}^8 \mathbb{B}_{aL}^t \mathbb{d}_a^{(e)} = \mathbb{B}_L^t \mathbb{d}^{(e)}.$$

Recall the expression for θ :

$$\theta_{ij} = \frac{1}{2} \left(\frac{\partial u_j}{\partial X_i} + \frac{\partial u_i}{\partial X_j} + \frac{\partial u_k^t}{\partial X_i} \frac{\partial u_k}{\partial X_j} + \frac{\partial u_k}{\partial X_i} \frac{\partial u_k^t}{\partial X_j} \right).$$

Note that there are two notations to represent spatial components in this thesis: (X, Y, Z) and (X_1, X_2, X_3) . There is no difference between these two notations. We decompose the matrix \mathbb{B}_L^t as

$$\mathbb{B}_L^t = \mathbb{B}^t + \mathbb{B}.$$

Similar to the derivation of Cauchy infinitesimal strain-displacement relation, we write out \mathbb{B}^t as:

$$\mathbb{B}^t = [\mathbb{B}_1^t, \mathbb{B}_2^t, \mathbb{B}_3^t, \mathbb{B}_4^t, \mathbb{B}_5^t, \mathbb{B}_6^t, \mathbb{B}_7^t, \mathbb{B}_8^t],$$

where

$$\mathbb{B}_a^t = \begin{bmatrix} \frac{\partial u_1^t}{\partial X} \frac{\partial N_a}{\partial X} & \frac{\partial u_2^t}{\partial X} \frac{\partial N_a}{\partial X} & \frac{\partial u_3^t}{\partial X} \frac{\partial N_a}{\partial X} \\ \frac{\partial u_1^t}{\partial Y} \frac{\partial N_a}{\partial Y} & \frac{\partial u_2^t}{\partial Y} \frac{\partial N_a}{\partial Y} & \frac{\partial u_3^t}{\partial Y} \frac{\partial N_a}{\partial Y} \\ \frac{\partial u_1^t}{\partial Z} \frac{\partial N_a}{\partial Z} & \frac{\partial u_2^t}{\partial Z} \frac{\partial N_a}{\partial Z} & \frac{\partial u_3^t}{\partial Z} \frac{\partial N_a}{\partial Z} \\ \frac{\partial u_1^t}{\partial X} \frac{\partial N_a}{\partial Y} + \frac{\partial u_1^t}{\partial Y} \frac{\partial N_a}{\partial X} & \frac{\partial u_2^t}{\partial X} \frac{\partial N_a}{\partial Y} + \frac{\partial u_2^t}{\partial Y} \frac{\partial N_a}{\partial X} & \frac{\partial u_3^t}{\partial X} \frac{\partial N_a}{\partial Y} + \frac{\partial u_3^t}{\partial Y} \frac{\partial N_a}{\partial X} \\ \frac{\partial u_1^t}{\partial Z} \frac{\partial N_a}{\partial Y} + \frac{\partial u_1^t}{\partial Y} \frac{\partial N_a}{\partial Z} & \frac{\partial u_2^t}{\partial Z} \frac{\partial N_a}{\partial Y} + \frac{\partial u_2^t}{\partial Y} \frac{\partial N_a}{\partial Z} & \frac{\partial u_3^t}{\partial Z} \frac{\partial N_a}{\partial Y} + \frac{\partial u_3^t}{\partial Y} \frac{\partial N_a}{\partial Z} \\ \frac{\partial u_1^t}{\partial Z} \frac{\partial N_a}{\partial X} + \frac{\partial u_1^t}{\partial X} \frac{\partial N_a}{\partial Z} & \frac{\partial u_2^t}{\partial Z} \frac{\partial N_a}{\partial X} + \frac{\partial u_2^t}{\partial X} \frac{\partial N_a}{\partial Z} & \frac{\partial u_3^t}{\partial Z} \frac{\partial N_a}{\partial X} + \frac{\partial u_3^t}{\partial X} \frac{\partial N_a}{\partial Z} \end{bmatrix}, \quad a = 1, \dots, 8.$$

As next step, we discretize the term involving nonlinear increments η_{ij} , $\int_{\Omega^e} S_{ij}^t \delta \eta_{ij} dX$. We construct matrices \mathbb{B}_{NL}^t and \mathbb{S}^t so that

$$\int_{\Omega^0} S_{ij}^t \delta \eta_{ij} dX \rightarrow (\delta \mathbb{d}^{(e)})^T \left(\int_{\Omega^0} (\mathbb{B}_{NL}^t)^T \mathbb{S} \mathbb{B}_{NL}^t dX \right) \mathbb{d}^{(e)}.$$

Thus, we construct \mathbb{S} as:

$$\mathbb{S} = \begin{bmatrix} S_{11}^t & S_{12}^t & S_{13}^t & 0 & 0 & 0 & 0 & 0 & 0 \\ S_{21}^t & S_{22}^t & S_{23}^t & 0 & 0 & 0 & 0 & 0 & 0 \\ S_{31}^t & S_{32}^t & S_{33}^t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & S_{11}^t & S_{12}^t & S_{13}^t & 0 & 0 & 0 \\ 0 & 0 & 0 & S_{21}^t & S_{22}^t & S_{23}^t & 0 & 0 & 0 \\ 0 & 0 & 0 & S_{31}^t & S_{32}^t & S_{33}^t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & S_{11}^t & S_{12}^t & S_{13}^t \\ 0 & 0 & 0 & 0 & 0 & 0 & S_{21}^t & S_{22}^t & S_{23}^t \\ 0 & 0 & 0 & 0 & 0 & 0 & S_{31}^t & S_{32}^t & S_{33}^t \end{bmatrix},$$

and \mathbb{B}_{NL}^t as:

$$\mathbb{B}_{NL}^t = [\mathbb{B}_{1NL}^t, \mathbb{B}_{2NL}^t, \mathbb{B}_{3NL}^t, \mathbb{B}_{4NL}^t, \mathbb{B}_{5NL}^t, \mathbb{B}_{6NL}^t, \mathbb{B}_{7NL}^t, \mathbb{B}_{8NL}^t],$$

where

$$\mathbb{B}_{aNL}^t = \begin{bmatrix} \frac{\partial N_a}{\partial X} & 0 & 0 \\ \frac{\partial N_a}{\partial Y} & 0 & 0 \\ \frac{\partial N_a}{\partial Z} & 0 & 0 \\ 0 & \frac{\partial N_a}{\partial X} & 0 \\ 0 & \frac{\partial N_a}{\partial Y} & 0 \\ 0 & \frac{\partial N_a}{\partial Z} & 0 \\ 0 & 0 & \frac{\partial N_a}{\partial X} \\ 0 & 0 & \frac{\partial N_a}{\partial Y} \\ 0 & 0 & \frac{\partial N_a}{\partial Z} \end{bmatrix}.$$

It can be verified that using designed matrices \mathbb{S} and \mathbb{B}_{NL}^t , $(\delta \mathbb{d}^{(e)})^T (\int_{\Omega^e} (\mathbb{B}_{NL}^t)^T \mathbb{S} \mathbb{B}_{NL}^t dX) \mathbb{d}^{(e)}$ is the discretized counterpart of the term $\int_{\Omega^e} S_{ij}^t \delta \eta_{ij} dX$.

Next, we look into the discretised version of the term $\int_{\Omega^e} S_{ij}^t \delta \theta_{ij}^t dX$. Just like the way we deal with the previous term, we can also construct a matrix \mathbb{S} , such that:

$$\int_{\Omega^e} S_{ij}^t \delta \theta_{ij}^t dX \rightarrow (\delta \mathbb{d}^{(e)})^T \left(\int_{\Omega^e} (\mathbb{B}_L^t)^T \mathbb{S} dX \right).$$

It is not hard to check that if we set

$$\mathbb{S} = \begin{bmatrix} S_{11}^t \\ S_{22}^t \\ S_{33}^t \\ S_{12}^t \\ S_{23}^t \\ S_{13}^t \end{bmatrix},$$

we can get the desirable result. Note that there are no coefficients “2” in front of the components S_{12}^t, S_{23}^t and S_{13}^t like in the strain tensor. The reason is that we have derived previously

$$\begin{bmatrix} \delta \theta_{11}^h \\ \delta \theta_{22}^h \\ \delta \theta_{33}^h \\ \delta 2\theta_{12}^h \\ \delta 2\theta_{23}^h \\ \delta 2\theta_{13}^h \end{bmatrix} = \mathbb{B}_L^t \delta \mathbb{d}^{(e)}.$$

Making use of the symmetry of the matrix S , we get the sum exactly by multiplying the vector $(\mathbb{B}_L^t \delta \mathbb{d}^{(e)})^T$ and \mathbb{S} . The only term left to be approximated is the external load term $\mathcal{R}_e^{t+\Delta t}$. This is related to the boundary term of the problem and will be discussed later.

To summarize, we get the discretized total Lagrangian formulation as:

$$\begin{aligned}
& (\delta \mathbb{d}^{(e)})^T \left(\int_{\Omega^e} (\mathbb{B}_L^t)^T \mathbb{C} \mathbb{B}_L^t dX \right) \mathbb{d}^{(e)} + (\delta \mathbb{d}^{(e)})^T \left(\int_{\Omega^e} (\mathbb{B}_{NL}^t)^T \mathbb{S} \mathbb{B}_{NL}^t dX \right) \mathbb{d}^{(e)} \\
& = \mathcal{R}_e^{t+\Delta t} - (\delta \mathbb{d}^{(e)})^T \left(\int_{\Omega^e} (\mathbb{B}_L^t)^T \mathbb{S} dX \right).
\end{aligned}$$

The above equation holds for any arbitrary virtual displacement. We can apply $(\delta \mathbb{d}^{(e)})^T = e_i, i = 1, \dots, 8$, to the equation in turn, where e_i is the base vector in \mathbb{R}^8 . Consequently, we get

$$\left(\int_{\Omega^e} (\mathbb{B}_L^t)^T \mathbb{C} \mathbb{B}_L^t dX \right) \mathbb{d}^{(e)} + \left(\int_{\Omega^e} (\mathbb{B}_{NL}^t)^T \mathbb{S} \mathbb{B}_{NL}^t dX \right) \mathbb{d}^{(e)} = \mathcal{R}_e^{t+\Delta t} - \left(\int_{\Omega^e} (\mathbb{B}_L^t)^T \mathbb{S} dX \right).$$

Let

$$\begin{aligned}
K_L^t &= \int_{\Omega^e} (\mathbb{B}_L^t)^T \mathbb{C} \mathbb{B}_L^t dX, \\
K_{NL}^t &= \int_{\Omega^e} (\mathbb{B}_{NL}^t)^T \mathbb{S} \mathbb{B}_{NL}^t dX, \\
F &= \int_{\Omega^e} (\mathbb{B}_L^t)^T \mathbb{S} dX.
\end{aligned}$$

To calculate these integrals, we need to transform them to the natural coordinate system as:

$$\begin{aligned}
K_L^t &= \int_{\Omega^e} (\mathbb{B}_L^t)^T \mathbb{C} \mathbb{B}_L^t dX = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 (\mathbb{B}_L^t)^T \mathbb{C} \mathbb{B}_L^t |J^{(e)}| d\xi d\eta d\zeta, \\
K_{NL}^t &= \int_{\Omega^e} (\mathbb{B}_{NL}^t)^T \mathbb{S} \mathbb{B}_{NL}^t dX = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 (\mathbb{B}_{NL}^t)^T \mathbb{S} \mathbb{B}_{NL}^t |J^{(e)}| d\xi d\eta d\zeta, \\
F &= \int_{\Omega^e} (\mathbb{B}_L^t)^T \mathbb{S} dX = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 (\mathbb{B}_L^t)^T \mathbb{S} |J^{(e)}| d\xi d\eta d\zeta,
\end{aligned}$$

where $|J^{(e)}|$ denotes the determinant of $J^{(e)}$.

These expressions show that the integrands contain rational algebraic functions in ξ, η, ζ . Only for the case that the element is a regular cube that the Jacobian matrix is constant. In these cases, the element integrals contain simple polynomials and the analytical expression can be derived. For general cases, the analytical integration in the natural coordinate system ξ, η, ζ is impossible and the best option is to use numerical integration.

Gaussian Integration

Gaussian quadrature rules are widely used in approximating the integral of a function, usually stated as a weighted sum of function values at specified points within the domain of integration. Here we use an 8-point ($2 \times 2 \times 2$) Gaussian integration in three dimensions. Gaussian rules for integrals in several dimensions are constructed by employing one-dimensional Gaussian rules on each coordinate axis separately.

To use the 2-point Gaussian quadrature rule in the domain $[-1, 1]$ to compute $\int_{-1}^1 g(\xi) d\xi$, the integration points within the interval are: $\xi_1 = \frac{-1}{\sqrt{3}}$, and $\xi_2 = \frac{1}{\sqrt{3}}$. The “weights” of the integration point are: $W_1 = 1$ and $W_2 = 1$. We get:

$$\int_{-1}^1 g(\xi) d\xi \cong g\left(-\frac{1}{\sqrt{3}}\right) + g\left(\frac{1}{\sqrt{3}}\right).$$

The remainder (difference from the exact value) is: $R = \frac{g^{(4)}(\bar{\xi})}{135}$, where $\bar{\xi}$ denotes a point in the interval $[-1,1]$. Thus, we achieve 4th order accuracy using this method. If the two-point rule is used in each coordinate axis to compute $\int_{-1}^1 \int_{-1}^1 \int_{-1}^1 g(\xi) d\xi d\eta d\zeta$, we get

$$\begin{aligned} \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 g(\xi) d\xi d\eta d\zeta &\cong \int_{-1}^1 \int_{-1}^1 \left\{ \sum_{l=1}^2 g(\xi_l, \eta, \zeta) W_l^1 \right\} d\eta d\zeta \cong \int_{-1}^1 \left\{ \sum_{m=1}^2 \sum_{l=1}^2 g(\xi_l, \eta_m, \zeta) W_l^1 W_m^2 \right\} d\zeta \\ &\cong \sum_{n=1}^2 \sum_{m=1}^2 \sum_{l=1}^2 g(\xi_l, \eta_m, \zeta_n) W_l^1 W_m^2 W_n^3. \end{aligned}$$

To write out the 8- point rule explicitly:

$$\begin{aligned} \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 g(\xi) d\xi d\eta d\zeta &\cong g\left(-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}\right) + g\left(\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}\right) \\ &+ g\left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}\right) + g\left(-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}\right) + g\left(-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right) \\ &+ g\left(\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right) + g\left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right) + g\left(-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right). \end{aligned}$$

Below is an illustration on the integration points for an 8-point Gaussian integration.

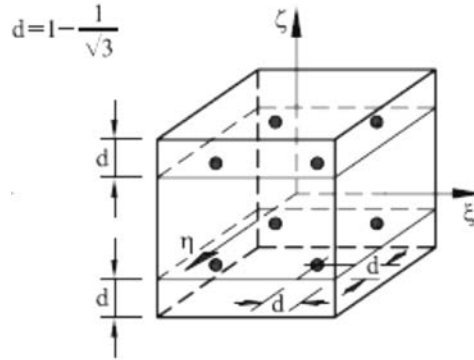


Figure 4.7 8-point Gaussian quadrature integration points [40]

Coming back to our computation of element stiffness matrices K_L^t, K_{NL}^t and F :

$$\begin{aligned} K_L^t &= \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 (\mathbb{B}_L^t)^T \mathbb{C} \mathbb{B}_L^t |J^{(e)}| d\xi d\eta d\zeta \\ &= \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 G_L^t d\xi d\eta d\zeta \cong G_L^t \left(-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}\right) + G_L^t \left(\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}\right) \\ &+ G_L^t \left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}\right) + G_L^t \left(-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}\right) + G_L^t \left(-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right) \\ &+ G_L^t \left(\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right) + G_L^t \left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right) + G_L^t \left(-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right), \end{aligned}$$

where

$$G_L^t = (\mathbb{B}_L^t)^T \mathbb{C} \mathbb{B}_L^t |J^{(e)}|.$$

Similar integrations can be done for the calculation of K_{NL}^t and F . Another common numerical integration rule is the Newton-Cotes rule, which will not be discussed in this thesis.

Discretized updated Lagrangian formulation

We proceed our discussion by discretizing the updated Lagrangian formulation. Due to many similarities with the discussion in the previous section, we will not repeat the calculation as far as it has been covered in the last section.

The updated Lagrangian formulation reads (applied on one element e)

$$\int_{\Omega_e^t} \frac{\partial S_{ij}^t}{\partial E_{rs}^t} \theta_{rs} \delta \theta_{ij} dx + \int_{\Omega_e^t} \sigma_{ij}^t \delta \eta_{ij} dx = \mathcal{R}_e^{t+\Delta t} - \int_{\Omega_e^t} \sigma_{ij}^t \delta \theta_{ij} dx.$$

We are going to discretize the above equation terms by terms.

Firstly,

$$\int_{\Omega_e^t} \frac{\partial S_{ij}^t}{\partial E_{rs}^t} \theta_{rs} \delta \theta_{ij} dx \rightarrow (\delta \mathbb{d}^{(e)})^T \left(\int_{\Omega_e^t} (\mathbb{B}_L^t)^T \mathbb{C} \mathbb{B}_L^t dx \right) \mathbb{d}^{(e)}.$$

The discretization proceeds pretty much the way it was discussed in the last section. Differences are:

- The integration is over the body configuration at time t instead of time 0.
- The Lagrangian elasticity tensor is derived at time t but with respect to the current configuration. This means that the second Piola–Kirchhoff stress tensor is calculated with respect to the current configuration and the linear incremental strain θ as well. Thus, \mathbb{B}_L^t is the same as the one in the last section except that its entries' partial derivatives taken with respect to the current Cartesian coordinates \tilde{x} .

Now we continue to discuss the second term:

$$\int_{\Omega_e^t} \sigma_{ij}^t \delta \eta_{ij} dx \rightarrow (\delta \mathbb{d}^{(e)})^T \left(\int_{\Omega_e^t} (\mathbb{B}_{NL}^t)^T \mathbb{S}^u \mathbb{B}_{NL}^t dx \right) \mathbb{d}^{(e)}.$$

Again, the right hand side term is very similar to the one we derived in the total Lagrangian case. Note that

$$\mathbb{S}^u = \begin{bmatrix} \sigma_{11}^t & \sigma_{12}^t & \sigma_{13}^t & 0 & 0 & 0 & 0 & 0 & 0 \\ \sigma_{21}^t & \sigma_{22}^t & \sigma_{23}^t & 0 & 0 & 0 & 0 & 0 & 0 \\ \sigma_{31}^t & \sigma_{32}^t & \sigma_{33}^t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{11}^t & \sigma_{12}^t & \sigma_{13}^t & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{21}^t & \sigma_{22}^t & \sigma_{23}^t & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{31}^t & \sigma_{32}^t & \sigma_{33}^t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{11}^t & \sigma_{12}^t & \sigma_{13}^t \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{21}^t & \sigma_{22}^t & \sigma_{23}^t \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{31}^t & \sigma_{32}^t & \sigma_{33}^t \end{bmatrix},$$

and the matrix \mathbb{B}_{NL}^t is the same as the one for the T.L. case except that its partial derivatives are taken with respect to the current Cartesian coordinates \vec{x} .

We turn to the last term:

$$\int_{\Omega^t} \sigma_{ij}^t \delta \theta_{ij} dx \rightarrow (\delta \mathbb{d}^{(e)})^T \left(\int_{\Omega^t} (\mathbb{B}_L^t)^T \mathbb{S}^u dx \right),$$

where

$$\mathbb{S}^u = \begin{bmatrix} \sigma_{11}^t \\ \sigma_{22}^t \\ \sigma_{33}^t \\ \sigma_{12}^t \\ \sigma_{23}^t \\ \sigma_{13}^t \end{bmatrix}.$$

Now, once again, we have derived the discretized equation:

$$\begin{aligned} (\delta \mathbb{d}^{(e)})^T \left(\int_{\Omega_e^t} (\mathbb{B}_L^t)^T \mathbb{C} \mathbb{B}_L^t dx \right) \mathbb{d}^{(e)} + (\delta \mathbb{d}^{(e)})^T \left(\int_{\Omega_e^t} (\mathbb{B}_{NL}^t)^T \mathbb{S}^u \mathbb{B}_{NL}^t dx \right) \mathbb{d}^{(e)} \\ = \mathcal{R}_e^{t+\Delta t} - (\delta \mathbb{d}^{(e)})^T \left(\int_{\Omega_e^t} (\mathbb{B}_L^t)^T \mathbb{S}^u dx \right). \end{aligned}$$

By applying $(\delta \mathbb{d}^{(e)})^T = e_i, i = 1, \dots, 8$, in turns, where e_i is the base vector in \mathbb{R}^8 , we obtain the final discretized equation for the unknown vector $\mathbb{d}^{(e)}$:

$$\left(\int_{\Omega_e^t} (\mathbb{B}_L^t)^T \mathbb{C} \mathbb{B}_L^t dx \right) \mathbb{d}^{(e)} + \left(\int_{\Omega_e^t} (\mathbb{B}_{NL}^t)^T \mathbb{S}^u \mathbb{B}_{NL}^t dx \right) \mathbb{d}^{(e)} = \mathcal{R}_e^{t+\Delta t} - \left(\int_{\Omega_e^t} (\mathbb{B}_L^t)^T \mathbb{S}^u dx \right).$$

Now, once again, we first transform these integrals to the natural coordinate system $\vec{\xi}$, and then we deploy the 8-point Gaussian integration rule to evaluate these integrals. The calculations go on exactly like that in the total Lagrangian case, so they will be omitted.

External work vector

We have discretized the continuous equations for both updated and total Lagrangian formulation. The only term left to be discretized is the term $\mathcal{R}_e^{t+\Delta t}$. We have mentioned in section 4.1.3 that it has the form:

$$\mathcal{R}_e^{t+\Delta t} = \sum_{k \in N_e} {}^k R_{t+\Delta t}^e \delta u_i = (\delta u)^T \sum_{k \in N_e} \mathbb{R}_{t+\Delta t}^k,$$

where $\mathbb{R}_{t+\Delta t}^k = [{}_1^k R_{t+\Delta t}^e, {}_2^k R_{t+\Delta t}^e, {}_3^k R_{t+\Delta t}^e]^T$ and N_e are the nodal sets in the element e that are connected to the springs:

$$\begin{aligned} {}_i^k R_{t+\Delta t}^e &= \mathcal{K} (D_i^{t+\Delta t} - u_i^{t+\Delta t}), & \text{if } k \in N_{II_a}, \\ {}_i^k R_{t+\Delta t}^e &= \mathcal{K} (0 - u_i^{t+\Delta t}), & \text{if } k \in N_{II_b}, \end{aligned}$$

$${}^k R_{t+\Delta t}^e = \mathcal{K}[(U_i^{t+\Delta t} - u_i^{t+\Delta t}) + (0 - u_i^{t+\Delta t})], \quad \text{if } k \in N_I,$$

for $i = 1, 2, 3$.

Using the displacement discretization $u^h(\tilde{\xi}) = \mathbb{N}\mathbb{d}^{(e)}$, we have

$$\delta u^h(\tilde{\xi}) = \mathbb{N} \delta \mathbb{d}^{(e)}.$$

Thus $\mathcal{R}_e^{t+\Delta t}$ is discretized as:

$$\mathcal{R}_e^{t+\Delta t} = (\delta \mathbb{d}^{(e)})^T \left(\mathbb{N}^T \sum_{k \in N_e} \mathbb{R}_{t+\Delta t}^k \right).$$

We notice that $\mathcal{R}_{t+\Delta t}^k$ contains the unknown variable $u_i^{t+\Delta t}$. We deal with this problem by making the following approximation:

$$(\mathcal{R}_e^{t+\Delta t})^{(k)} \cong (\delta \mathbb{d}^{(e)})^T \left(\mathbb{N}^T \sum_{k \in N_e} (\mathbb{R}_{t+\Delta t}^k)^{(k-1)} \right),$$

where $(\mathcal{R}_e^{t+\Delta t})^{(k)}$ denotes the value of $\mathcal{R}_e^{t+\Delta t}$ at the k th iteration in the Newton-Raphson procedure computing from time step t to $t + \Delta t$. $(\mathbb{R}_{t+\Delta t}^k)^{(k-1)}$ denotes the value of $\mathbb{R}_{t+\Delta t}^k$ computing from $u^{(t+\Delta t)(k-1)}$, where $u^{(t+\Delta t)(k-1)}$ denotes the value of $u^{t+\Delta t}$ at the $(k-1)$ th iteration computing $u_{t+\Delta t}$ from time t to $t + \Delta t$. To summarize, we derive the external work vector as

$$\mathbb{R}_e^{t+\Delta t} = \mathbb{N}^T \sum_{k \in N_e} (\mathbb{R}_{t+\Delta t}^k)^{(k-1)}.$$

Element equations assemblage to obtain the overall equilibrium equations

In the last section we discretized the linearized continuous equation term by term within an element. The global matrices or vectors are obtained by adding up the contributions from every element.

In this section we will assemble the element matrices/vectors to derive the global discretized system. Note that this step is implemented in Marc with a default assemblage procedure. Hence, the user has no freedom in it. I will only explain the general idea of element equations assemblage. To check how Marc assembles the element equations, see [35].

Note that the interpolation functions defined in section “Selection of a proper interpolation function” have compact supports. For a particular element, only the shape functions defined on it, namely, N_a , $a = 1, \dots, 8$, will be nonzero. Thus we can expect a lot of zeros in the element matrices. We denote the element matrix which only consists of the components related with the shape functions of the nodes within the element as $[\mathbb{K}_{pq}]$. The assembly process is all about putting the non-zero components of the element matrices into the right position in the global matrices. To assemble the matrices, the following arrays are needed.

Firstly, we need to set up the local matrix array LM, which relates the local node number, degree of freedom and element number with the global equation number as,

$$LM(i, a, e) = ID(i, IEN(a, e)),$$

where i denotes the degree of freedom, which is 3 for a 3D displacement-based element, a is the local node number, and e represents the element number. ID mapping maps the global node numbers to global equation numbers. IEN represents the element nodes array, which relates local node numbers to global node numbers as:

$$IEN(a, e) = A,$$

where A is the global node number.

Hence, for example, if we have derived the element matrices, to find out where to add the position of the components k_{pq} in the global matrix, we need to first find out how this component related with the local nodes through the relation:

$$\begin{aligned} p &= 3(a - 1) + i, \\ q &= 3(b - 1) + j, \end{aligned}$$

where $1 \leq i, j \leq 3$.

Using these relations we can find the related local nodes a and b . We can search for the corresponding global nodes numbers A and B given the element number e and local node numbers a and b . Finally, using the ID matrix, we find the equation number in the global system:

$$\begin{aligned} P &= ID(i, A), \\ Q &= ID(j, B). \end{aligned}$$

Then the addition process goes on as:

for all the element matrices, $[k_{pq}]$, for every element in $[e]$,

$$K_{PQ} = K_{PQ} + k_{pq}.$$

The vectors assembled in the same way as matrices.

Solution for the unknown nodal displacements

Basically after we assembled the element matrices in the last section, we get the global discretized linearized system (for each iterative step):

$$K\Delta U = R - F,$$

where K is the global tangent stiffness matrix, ΔU is the incremental displacement at all nodes, R is the global external work vector and F is the internal work vector. Below displays a flow graph illustrating how Newton-Raphson method is deployed to compute the unknown displacement field U from time t to $t + \Delta t$. Note that the subscript indicates the time and the superscript indicates the iteration step within the time step.

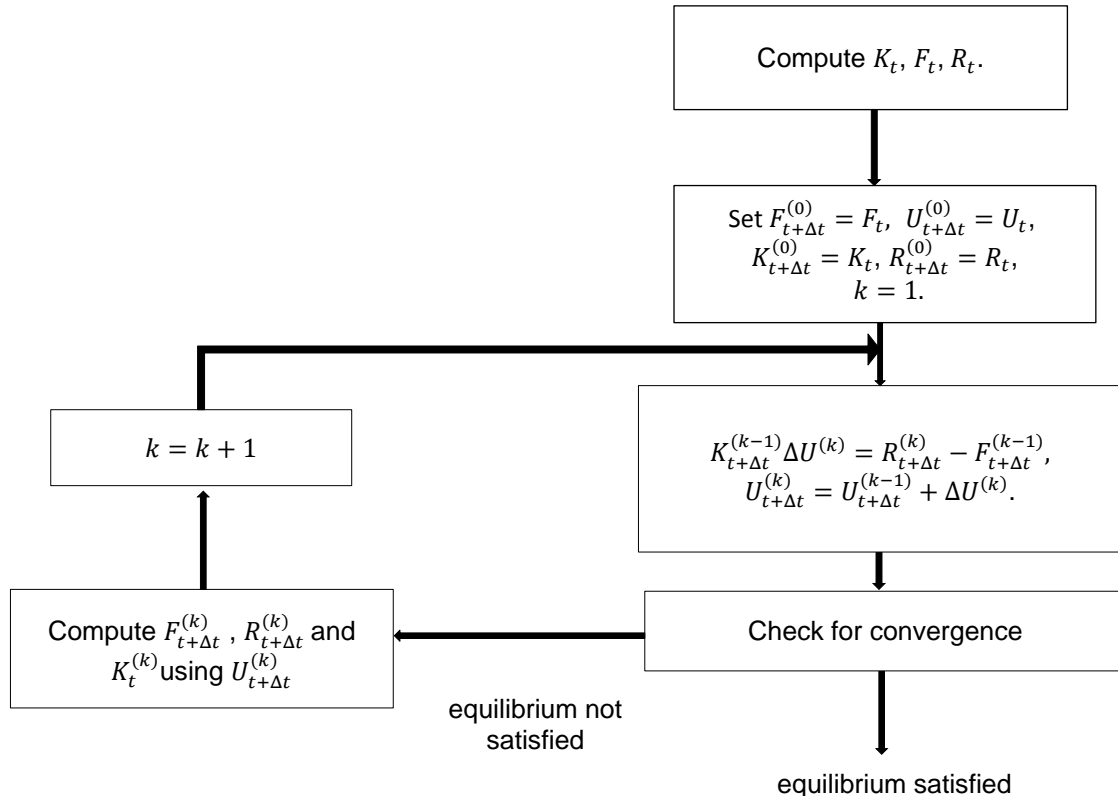


Figure 4.8 Newton-Raphson solution flow graph

Below are several comments on the graph:

- Note that in section 4.1.4, we have mentioned the approximation we made in computing $R_{t+\Delta t}^{(k)}$, namely

$$R_{t+\Delta t}^{(k)} \rightarrow R_{t+\Delta t}^{(k-1)}.$$

Thus the equation

$$K_{t+\Delta t}^{(k-1)} \Delta U^{(k)} = R_{t+\Delta t}^{(k)} - F_{t+\Delta t}^{(k-1)},$$

actually is calculated as

$$K_{t+\Delta t}^{(k-1)} \Delta U^{(k)} = R_{t+\Delta t}^{(k-1)} - F_{t+\Delta t}^{(k-1)}.$$

- The convergence criteria we deployed here is: residual checking, namely,

$$\frac{\|F_{residual}\|_{\infty}}{\|F_{reaction}\|_{\infty}} < \text{TOL},$$

where $F_{residual} = R - F$ and $F_{reaction} = R$, TOL is the control tolerance.

4.2. Morphing method (a computer vision-based method)

In this project, besides FEM-based models, a morphing method is also deployed parallel to open the mouths virtually. Image morphing is a popular class of techniques for producing transitions between images which is based on interpolating the positions and colours of pixels in two images. For our application, the morphing technique is applied on the range images (depth maps).

A **depth map** is by essence a matrix whose elements store the depths of points in certain view port. Imagine that we want to represent a surface in 3D. Then we need to specify the coordinates in x, y, z -directions of each point on the surface. Now we let the view port discretize the x, y -plane. For example, if we specify a view port as $[-150, 150]$ in both x and y directions, and 1 as the step of the grid (Note that this is also the view port we use to show all the depth maps in this thesis with the unit $1mm$), we will end up with 301×301 uniformly distributed grid points in the x, y -plane, whose x, y -coordinates read $(i, j), i = -150, \dots, 150, j = -150, \dots, 150$. The depth map, as its name suggests, assigns the z -coordinates of each grid point according to its location on the x, y -plane. Therefore, the depth map will be a 301×301 matrix. If we let Z_{buffer} denote the matrix, $Z_{buffer}(i, j)$ equals the value of the z -coordinate of the point on the surface whose x, y -coordinates are i and j . This way, combining the view port and the depth map, we are able to describe a (discretized) surface.

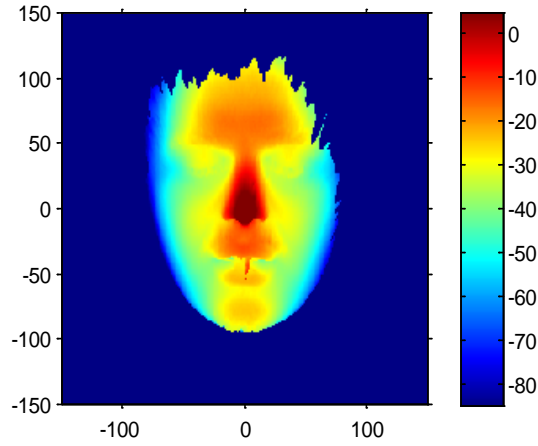


Figure 4.9 Depth map of a human face

Figure 4.9 is a typical depth map of an open mouth image. Given a depth map and view port, such images can be rendered by MATLAB. From the image we can see a clear human face shape (non-dark blue area), this is due to the fact that we set the background depth as -10^6 , while the human face depth values are varying from -80 to -5 . We can also observe the features of a human face due to their difference in depth. For example, the nose tip is the most protruding point of a face (hence has the largest z -coordinate value indicated by dark red colour) and the eyebrow and forehead locate more outward than the eyes.

A morphing is determined from two 3D surfaces (depth maps) \mathcal{I}_0 and \mathcal{I}_1 and maps $C: \mathcal{I}_0 \rightarrow \mathcal{I}_1$ specifying a complete one-to-one correspondence between points in the two surfaces. In practice, C is partially specified on a sparse set of matching points in the two 3D models:

$$P = \{p_1, \dots, p_N\} \quad \text{and} \quad Q = \{q_1, \dots, q_N\}$$

where

$$C(p_i) = q_i \quad \text{for } i = 1, \dots, N.$$

The remaining correspondences are determined by interpolation

$$C(p) - p = \frac{1}{\sum_{k=1}^n w_k(p)} \sum_{k=1}^n w_k(p) (C(p_{N(k)}) - p_{N(k)}),$$

where $N(1), \dots, N(n)$ are $n = 3$ neighbours of p , $w_k(p)$ are defined inverse proportional to the distance between p and $p_{N(k)}$

$$w_k(p) = \frac{1}{\varepsilon + \|p - p_{N(k)}\|},$$

where $\varepsilon = 0.00001$ is introduced to prevent division by zero when p coincides with one of the p_1, \dots, p_N . The set of neighbours is uniquely determined by imposing a triangulation on points p_1, \dots, p_N . Then, for any p , it belongs to at least one triangle. The vertices of this triangle are defined as neighbours of p . For points outside of the triangulation grid the neighbours of p are defined as three closest points from the set p_1, \dots, p_N . Thus for any point of the model p we can determine its local coordinates of the triangulation grid, so called *3D Mesh coordinates*:

$$p \rightarrow (w_1(p), w_2(p), \dots, w_n(p), N(1), \dots, N(n), R)$$

where

$$p = \frac{1}{\sum_{k=1}^n w_k(p)} \sum_{k=1}^n w_k(p) p_{N(k)} + R.$$

In the Mask Design Tool the correspondence points $P = \{p_1, \dots, p_N\}$ are generated automatically from the coordinates of 10 Principal Landmarks with indices: P30, P20, P2, P10, P11, P3, P27, P14, P15, P6, as shown on Figure 4.10 (Left). Using some linear combinations between X- and Y- coordinates of the principal landmarks, another set of 32 auxiliary landmarks are defined. Thus the positions of the auxiliary landmarks are scaled together with the principal landmarks. In total 40 landmarks are triangulated into a 3D mesh grid with 56 triangles, see Figure 4.10 (Right).

Typically, the morphing of Scan1 (closed mouth scan) to Scan2 (open mouth scan) consists of the following steps:

1. Automatically detect or manually annotate the set of principal 10 landmarks on Scan1 and Scan2.
2. Compute 3D Mesh grids for both scans.
3. Find 3D Mesh grid coordinates for all vertices of Scan 1, i.e.

$$(w_1(p), w_2(p), \dots, w_n(p), N(1), \dots, N(n), R)$$

4. Apply the 3D Mesh grid coordinates to 3D Mesh of Scan 2 to find the morphed positions:

$$C(p) = \frac{1}{\sum_{k=1}^n w_k(p)} \sum_{k=1}^n w_k(p) C(p_{N(k)}) + R.$$

5. Generate the new Zbuffer, if required.

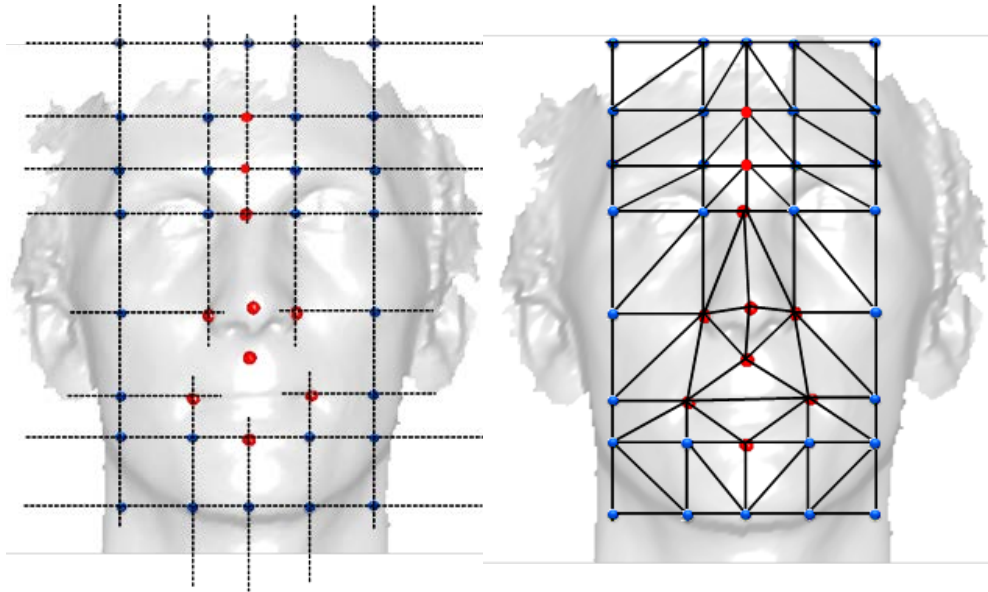


Figure 4.10 Landmarks -> 3D mesh. Left: 10 Principal landmarks are shown as red balls, 32 auxiliary as blue balls. Right: imposed triangulation grid.

The steps we followed to virtually open the mouth are:

1. Set open mouth landmarks and read closed mouth scan landmarks.
2. Generate two 3D mesh grids for both closed and open mouth cases according to their landmark positions respectively, see Figure 4.11.
3. Get 3D mesh grid point coordinates from the 3D mesh for closed mouth and open mouth respectively.
4. Derive the transformation between the closed mouth and open mouth 3D mesh grid coordinates based on the coordinates we derived in the last step.
5. Linearly interpolate all the points within the triangles (the 3D mesh grid).
6. Render the morphed (open-mouth) image using the coordinate and connectivity information.

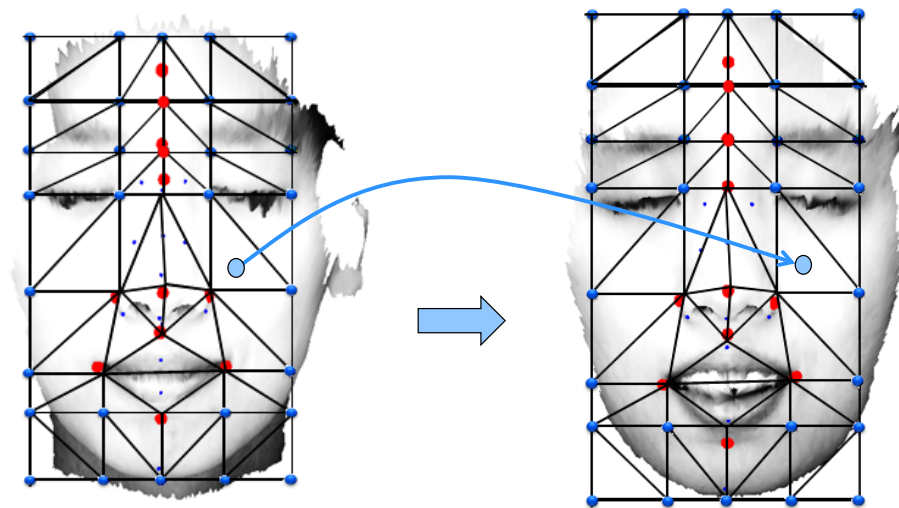


Figure 4.11 Mouth opening morphing transformation

4.3. Summary of methods

Having presented all the ingredients for our mouth opening tool, the following table provides an overview about the methods we developed in this project.

Table 2 Methods overview

<i>Method Shorthand</i>	<i>MORPHING</i>	<i>FEM I</i>	<i>FEM II</i>	<i>FEM III</i>	<i>FEM IV</i>
Physically-based model	No	Yes	Yes	Yes	Yes
Complexity	Low	High	High	High	High
Element type	N/A	Shell element	Hexahedral element	Hexahedral element (Herrmann formulation)	Hexahedral element (Herrmann formulation)
Material model	N/A	Purely elastic	Purely elastic	Mooney	Gent
Young's Modulus:	N/A	0.5	0.5	N/A	N/A
Poisson's ratio:	N/A	0.42	0.42	N/A	N/A
Spring stiffness	N/A	1	1	0.001	1
Hyperelastic material constants	N/A	N/A	N/A	$C_{10} = 0.0094$ $C_{01} = 0.082$	$E = 0.75$ $I_m = 4$
Compressibility	N/A	Compressible	Compressible	Nearly incompressible	Nearly incompressible
Formulation	N/A	U.L.	T.L.&U.L.	T.L.	U.L.&T.L.

4.4. Implementation details

In this section, we will elaborate the implementation procedure of FEM-based model simulation. Generally speaking, the FEM-based models are implemented using both MATLAB (in Windows and Linux systems) and Marc (in Linux operating system). Marc is a general-purpose, nonlinear finite element modelling tool. For more information about Marc, check [41]. Shortly speaking, the implementation includes: firstly, preparation for the input file for Marc using MATLAB (obtaining the close mouth scan from the scan database, computing mouth opening parameter, etc.), then, finite element analysis carried out by Marc based on the input file, finally, result-retaining from Marc and open mouth image generation using MATLAB. There are 11 steps involved in our open mouth tool implementation procedure:

Step 1. Pre-selection of the closed and open mouth pairs from the database (only needed for model-developing)

Philips has a database of various facial 3D scans. We need to select “good” scan pairs (of open and closed mouth scans) for development and evaluation of the soft tissue models. Our “good” scan pair satisfies the following conditions: firstly, both open and closed mouth scans belong to the same person; secondly, the scan pair can be successfully aligned to each other in the upper face area where the open and closed mouth faces are basically the same. The alignment is done with a version of Iterative Closed Point algorithm implemented at Philips. Figure 4.12 and Figure 4.13 present an example of a well-aligned and a badly-aligned scan pairs by means of the difference depth maps.

The difference maps are obtained by subtracting the closed mouth depth map from the open mouth depth map. The result indicates the difference of two surfaces in the z -direction.

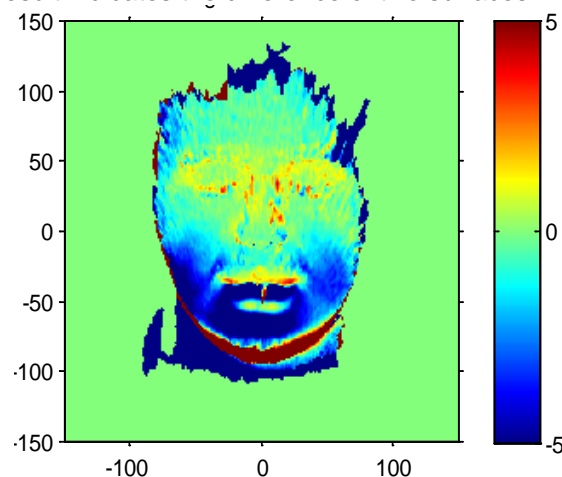


Figure 4.12 Example: Well-aligned upper face

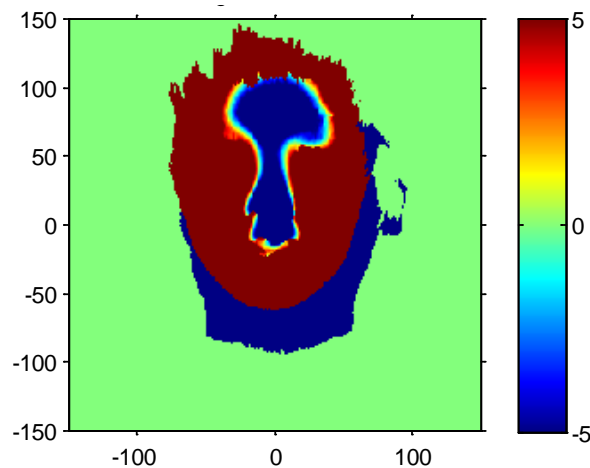


Figure 4.13 Example: Badly-aligned upper faces

We can see that for the properly aligned scans the difference in the upper head is close to zero, (indicated with green colour), while on the lower part of the face the difference is negative because of jaw moving down-backwards (indicated with dark blue colour). The badly aligned pairs of scans can be visually identified by the extreme difference values across the face. There are various reasons that might cause two scans cannot be aligned at the forehead part. The most probable reasons are the 3D scan artefacts and the large variation in the scanning angle.

Step 2. Extract and edit related information (landmarks) from database

The scans are stored in the database together with the list of coordinates of facial landmarks. Landmarks play an important role in our models. By landmarks, we mean distinctive geometrical feature points on the human face which are easy to identify on the face like mouth and eye corners, nose tip, etc. Philips has developed its own landmark annotation techniques, via which landmarks on facial scans will be automatically detected. Since these tools are not perfect Philips also developed a GUI by which the landmarks positions can be manually annotated and refined. Figure 4.14, Figure 4.15 and Figure 4.16 are the snapshots of the landmarks annotation GUI, using which one can check all the 43 landmarks (blue dots) including 10 prime landmarks (red dots).

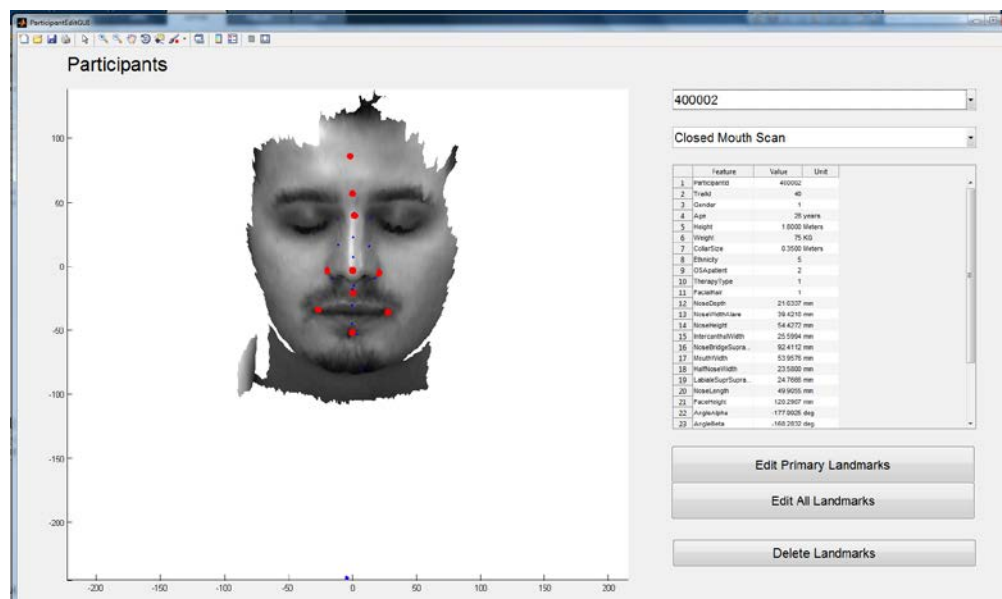


Figure 4.14 Landmarks annotation GUI

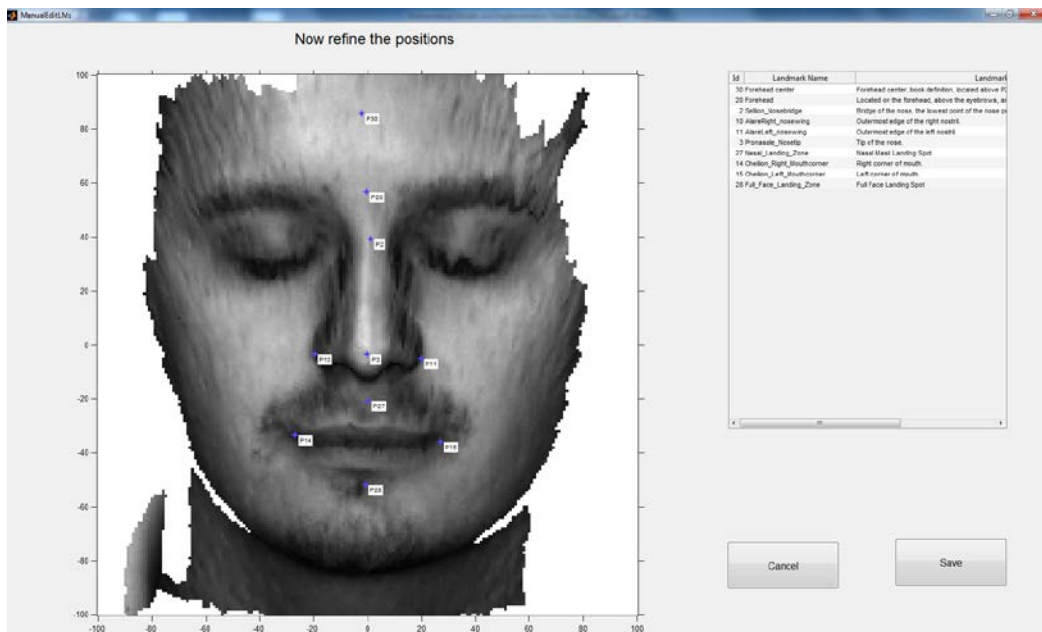


Figure 4.15 Edit prime landmarks window

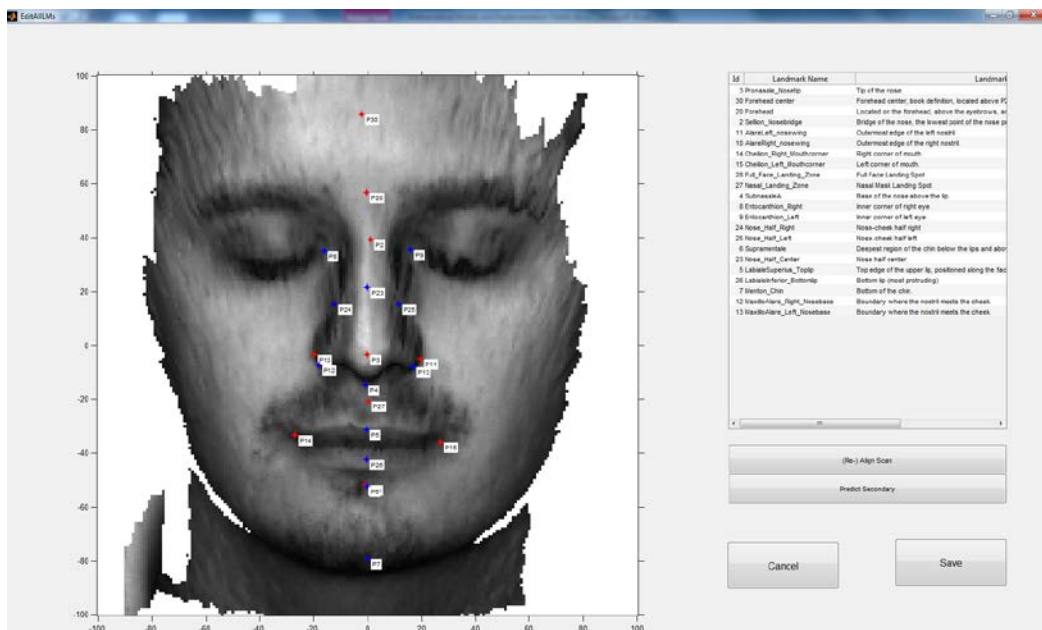


Figure 4.16 Edit all landmarks window

Currently, there are no landmarks automatically added for the open mouth scans in the database. Figure 4.17 shows a sample open mouth scan in the GUI. The “good” pairs of the open-closed mouth scans the landmarks can be copied from the closed mouth scan as explained in the following step. Whenever needed, one can manually correct the landmark positions in the GUI.

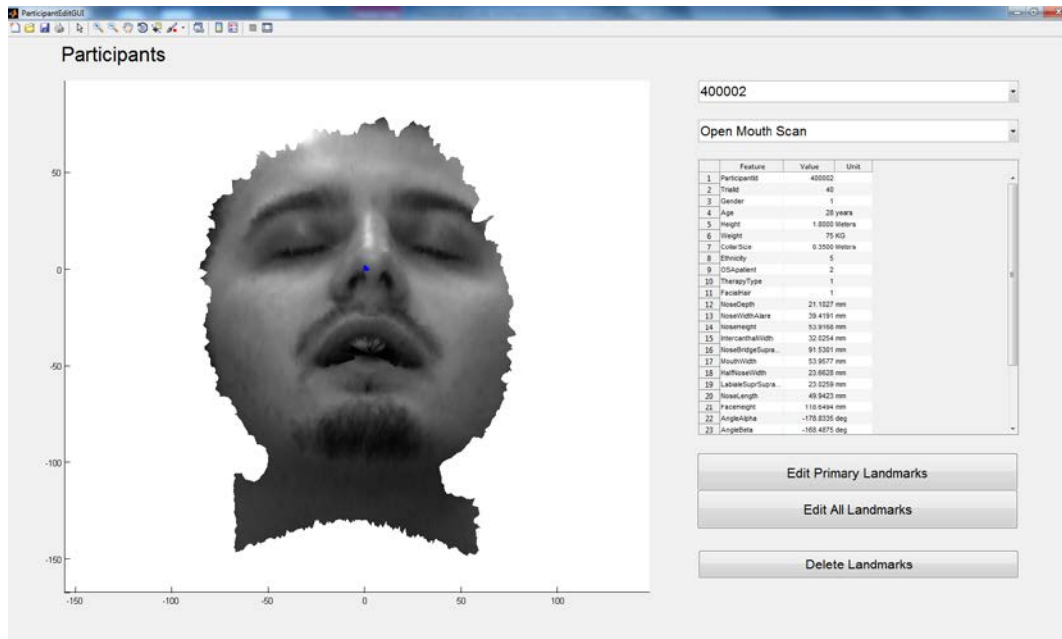


Figure 4.17 Open mouth case (without landmarks)

Step3. Specify the translation and rotation parameter for the mouth open motion

Our simulations uses four parameters (t_1, t_2, t_3, r), to describe the rigid movement of the jaw. t_1, t_2 and t_3 are used to specify the translation of the lower skull in x, y, z -direction respectively and r (in radians) gives the rotation angle of the jaw around the horizontal axis. The four rotation-translation parameters are saved in the parameter file, which is then read by Marc procedures.

The major goal of the assignment is to open the mouth uniformly on all scans with identical rotation-translation parameters (for example specified in the mouth opening tool). At the same time, in order to correlate the FEM models with the ground truth data, one have to know for each closed-open scan pair in the test set, the specific rotation-translation parameters.

These parameters are estimated automatically by means of a patch matching algorithm. First, we use the Iterative Closest Point (ICP) alignment to match the nose bridge area of the open mouth scan to the nose bridge area of the closed mouth scan. This alignment gives also the alignment of the upper skulls which are one-to-one related with the facial surfaces. Next, the ICP alignment is used again to match the chin area of the closed mouth scan to the chin area of the aligned open mouth scan. This alignment gives also the alignment of the lower skulls (jaws) which are also one-to-one related with the facial surfaces. The second alignment gives the rotation translation parameters of the jaw's movement. Moreover, the alignments allow to copy the facial landmarks from the closed mouth scan to the open mouth scan where the upper face landmarks are copied after the first alignment, the chin landmarks are copied after the second alignment and the mouth corner landmarks are defined as the average of the corner landmark positions at the first and the second alignments.

We should point out that the movement of the jaw during mouth opening is much more complicated than we assumed in this model (the moving jaw) which might be in reality non-linear. Figure 4.18 and Figure 4.19 illustrate the procedure described above.

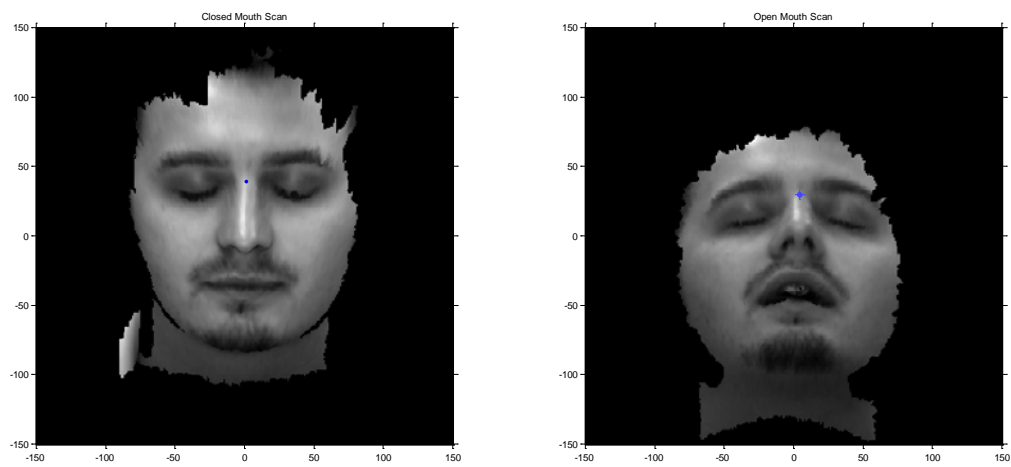


Figure 4.18 Before any alignment

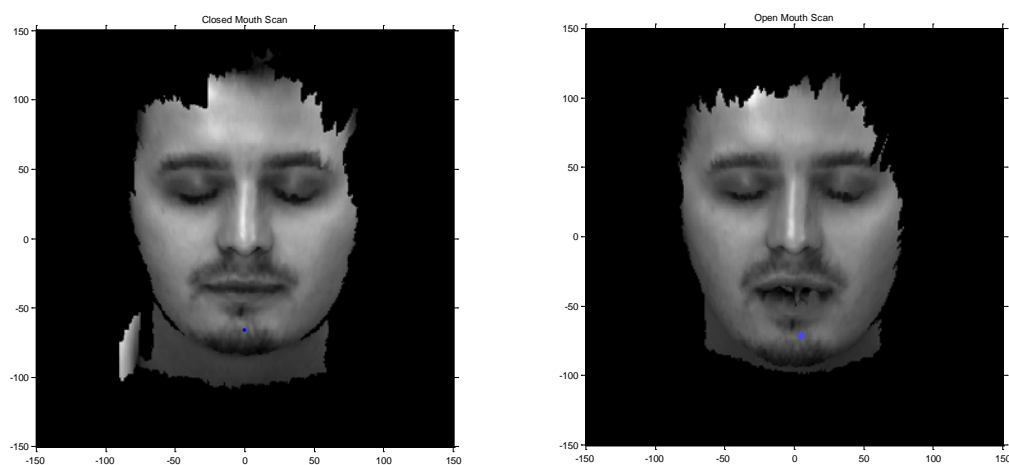


Figure 4.19 After first (upper head) alignment

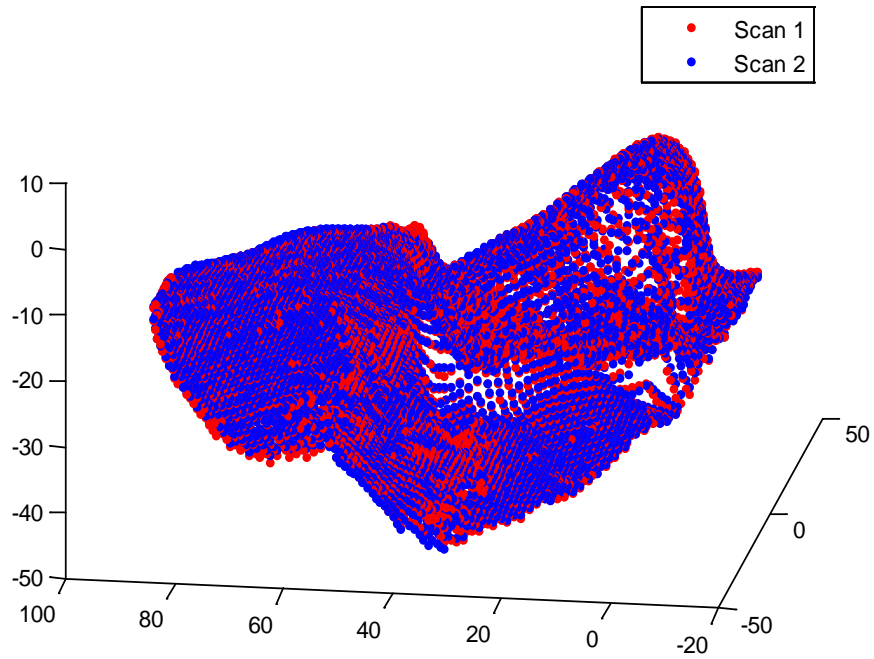


Figure 4.20 First alignment result 3D plot (Scan1: aligned closed mouth scan, Scan 2: open mouth scan)

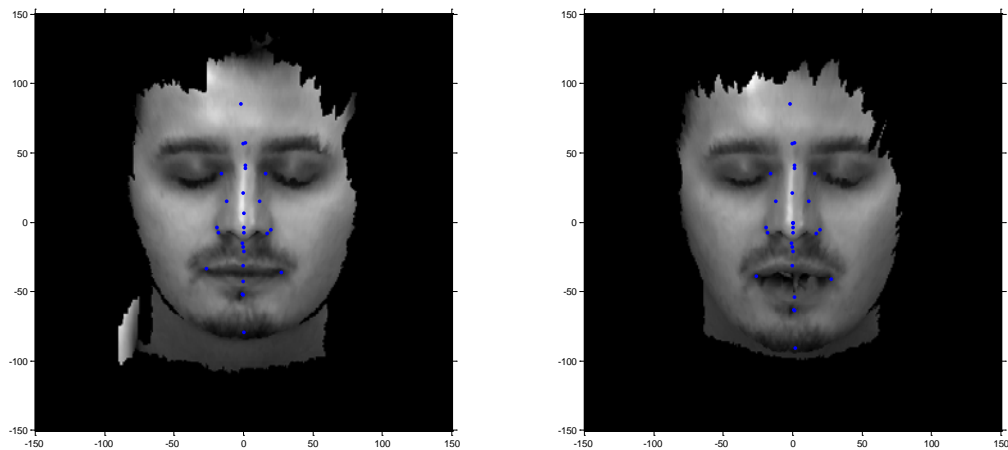


Figure 4.21 After second (jaw) alignment with landmarks computed for open mouth scan

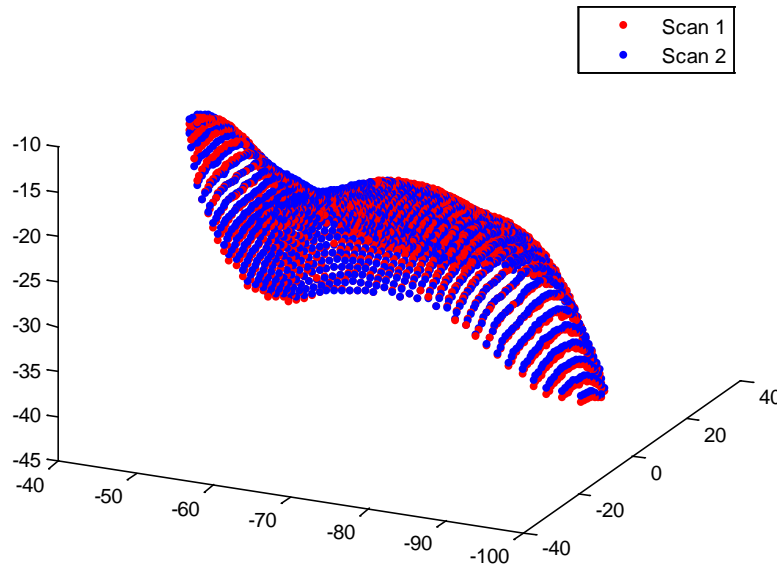


Figure 4.22 Second alignment result 3D plot (Scan1: aligned closed mouth scan; Scan 2: open mouth scan)

Step 4. Obtain skin thickness information (morphing technique)

As part of input information of our models, local skin thickness is of importance in terms of realism of our models. There are two skin thickness maps (matrices whose entries store the thickness of the soft tissue according to its location on the x, y -plane) available at Philips for an “averaged face” geometry. The “averaged face” was computed by averaging and morphing a number of facial scans. The first soft tissue thickness map is created as an interpolation of 20 average thicknesses obtained via literature study. The second one is derived from MRI data by Philips interneer, Mingming [42]. For our FEM simulations we use the first map which has more realistic values on the forehead and on the nose bridge.

Features of a face have different proportions and shapes for different individuals. For example, the width of the mouth, the shape of the nose or more generally, the different distance of specific features. As a whole, each face has difference in length and width as well. Given all differences mentioned above, the thickness map for an averaged face should be morphed to match a specific face before it is applied. The morphing procedure is outlined below.

First, we align the specific closed mouth face to the averaged face to minimize the RMS (Root Mean Square) distance between the corresponding landmarks. Next, we generate two 3D grids based on the landmark positions of the average and of the particular face, like illustrated in Figure 4.10.

The next step is to get the mesh-coordinates for all vertices of the particular face. Using the mesh-coordinates we find the corresponding points on the average face and sample the soft tissue thickness. Thus we obtain the soft tissue thickness for all vertices of the particular face.

Step 5. Generate skin geometry

All FEM models derived in this work consist of elements defined on the regular grid. In order to define a simple yet accurate skin geometry the 3D model, the depth map of the closed mouth face is rendered on a $3mm$ grid. Then the “mouth-cut” is made to let the model be able to open its mouth. Specifically, according to the landmark positions of mouth, another set of mouth nodes is created with $0.5mm$ extrusion in the z -direction. The extrusion is needed to facilitate the separation of the two sets of nodes in Marc Mentat software. The connectivity of the nodes is then adapted to append them to the surface so that the mouth can open realistically. This can be illustrated as below:

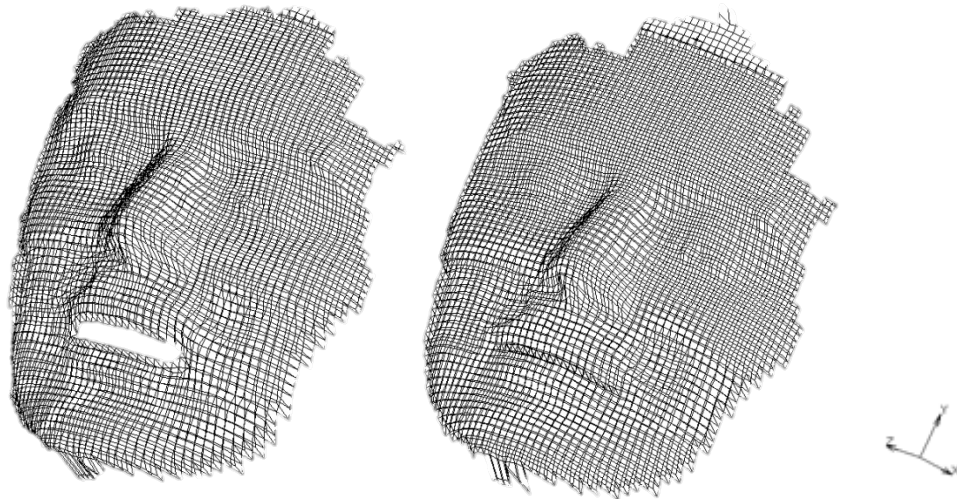


Figure 4.23 Skin geometry open (left) and closed (right) mouth

Step 6. Generate skull geometry

The soft tissue thicknesses are rendered on the same $3mm$ grid where the values in-between the 3D model vertices are obtained via linear interpolation. The skull geometry is generated by subtracting the rendered soft tissue thickness map from the rendered surface map. We define the lower skull according to the landmarks which indicate the height of the mouth. We simplify the jaw geometry by cutting the skull horizontally at the average mouth corner position, as shown in Figure 4.24:

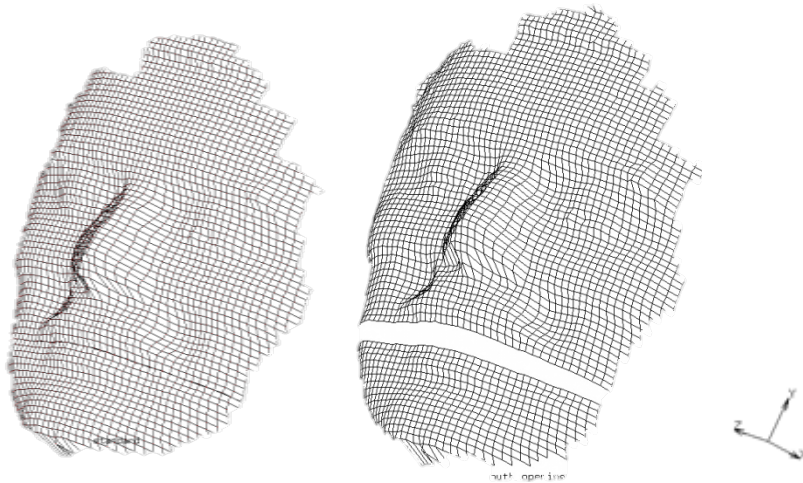


Figure 4.24 Skull geometry open (right) and closed (left) mouth

Step 7. Assemble skin and skull geometry

After we derive skin and skull geometries (namely, node coordinates and connectivity), it remains to assemble them. Content wise, we first remove the background grid nodes around the face where either skin or skull values are missing. This allows reducing of the computational burden. Second, we connect the nodes into the elements to get a one-to-one mapping between skin surface nodes and skull surface nodes except from the mouth side area where by design we have one skin node connected to two skull nodes.

Step 8. Complete the input file for Marc

In this step, we complete the input file for Marc by adding spring connections between skin layer and skull layer and other parameters. It includes specifying built-in parameters, like the element type we use, material properties and everything needed under the proprietary Marc input file format, see Appendix 0.

Step 9. Run the simulation in Marc

With the complete input file, the simulation is ready to run. We make a MATLAB function in Linux environment which can detect the existence of the input file, see Appendix A.2. As a result, once the completed input file is present in the specified folder, Marc gets started. The operation of the Marc is controlled by means of procedure file, see Appendix A.3. Marc interprets the commands in the procedure file and assigns the proper boundary conditions of the model (the prescribed position of the jaw skull and the clamped upper skull) and computes the resulting open-mouth geometry.

Step 10. Process the output file from Marc

At the end of the simulation, a user subroutine is automatically executed to produce the simulated open mouth skin surface nodes coordinates. Note that along this coordinate file, there are also some other files generated by Marc which can be used to visualize the result in Marc. Some files (especially .log and .out file) are very useful in debugging of the model and the procedure files. Based on the newly generated coordinates, the open mouth depth map can be rendered back in MATLAB in Windows system. Figure 4.25 shows the simulated result (left) and the ground truth scan (right) for a specific participant.

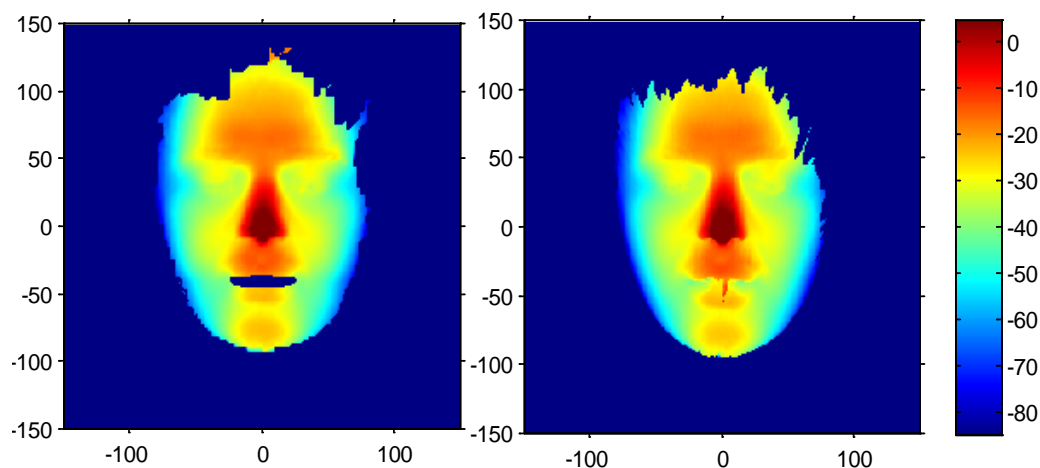


Figure 4.25 Simulation result (left) and ground truth scan (right)

Step 11 Post-processing

We perform the mouth open simulation on 62 closed-open mouth scan pairs to evaluate the performance of different methods. After some calculation, described in detail in section 5.2, we are able to obtain an average and the standard deviation error map for each method on the average 3D face model. The average error map can be used for systematic correction of the open mouth scans after the morphing procedure. Note that the average error map is base only on 62 faces and therefore it is only an approximation to the real systematic error of each method. In section 5.2 we will explain how the standard deviation map can indicate how close the average error map is to the real systematic error.

5. Results and discussion

In this chapter, we present the benchmarking of different methods. We implement the procedure described in section 4.4 (FEM-based methods) and the morphing method for 62 participants who have both open and closed mouth scans stored in the database. As a result, we constructed 62×5 depth maps (Z_{buffer_S}) which store the simulated open mouth geometry on a specified view port for each individual and for each method. On the other hand, original depth maps which store the real life open mouth geometry are also available to be used for verification purpose. The depth maps derived from the original open mouth scan will be referred as ground truth ($Z_{\text{buffer}_{GT}}$). We point out once again that the view port we use is a square window (from -150mm to 150mm) both in x and y directions sampled with the grid step (resolution) of 1mm . Thus the dimension of the depth map matrices is 301×301 . The difference map (Z_{buffer_D}) is defined as:

$$Z_{\text{buffer}_D} \stackrel{\text{def}}{=} Z_{\text{buffer}_{GT}} - Z_{\text{buffer}_S}.$$

Obviously, Z_{buffer_D} gives a direct indication of the difference in the surfaces in z -direction and therefore it is used as an indication of simulation error for individual scans and methods.

To derive a quantitative indication on how accurate each method is, simple statistical methods are applied. We compute the following statistical error indicators: RMS (Root Mean Square) error and average error and standard deviation map. In this chapter, the simulation results of the five methods (FEMI, FEMII, FEMIII, FEMIV and MORPHING) will be compared in a comprehensive way.

5.1. Benchmarking I: RMS error

In this section, we first present the computation procedure of RMS error of an individual difference map (Z_{buffer_D}). Then we show the RMS results of 62 individuals for FEMII and MORPHING in a figure. Finally, we average the individual results (over 62 participants) for each methods to derive a RMS error indicator for each method. The results are then summarized in a table.

The individual RMS error is computed as follows:

1. The difference map (Z_{buffer_D}) contains two types of errors: 1) errors in the lower part of the face around cheeks and chin due to the inaccuracy of the mouth opening method (modelling error); 2) errors on the boundary edges of the depth maps due to different size of the scanned areas and heads. The second type errors are not relevant for estimation of the performance, and should be ignored. Since these errors have typically very high amplitude, they can be detected by simple comparison with a (fixed) threshold. We assume the modelling error does not exceeds 5 mm . If the absolute value of entries of Z_{buffer_D} larger than 5 mm , we set it to 0, written in formula as:

$$\text{if } |Z_{\text{buffer}_D}|_{ij} \geq 5 \Rightarrow (Z_{\text{buffer}_D}')_{ij} = 0, \quad \text{for } i, j = 1, \dots, 301,$$

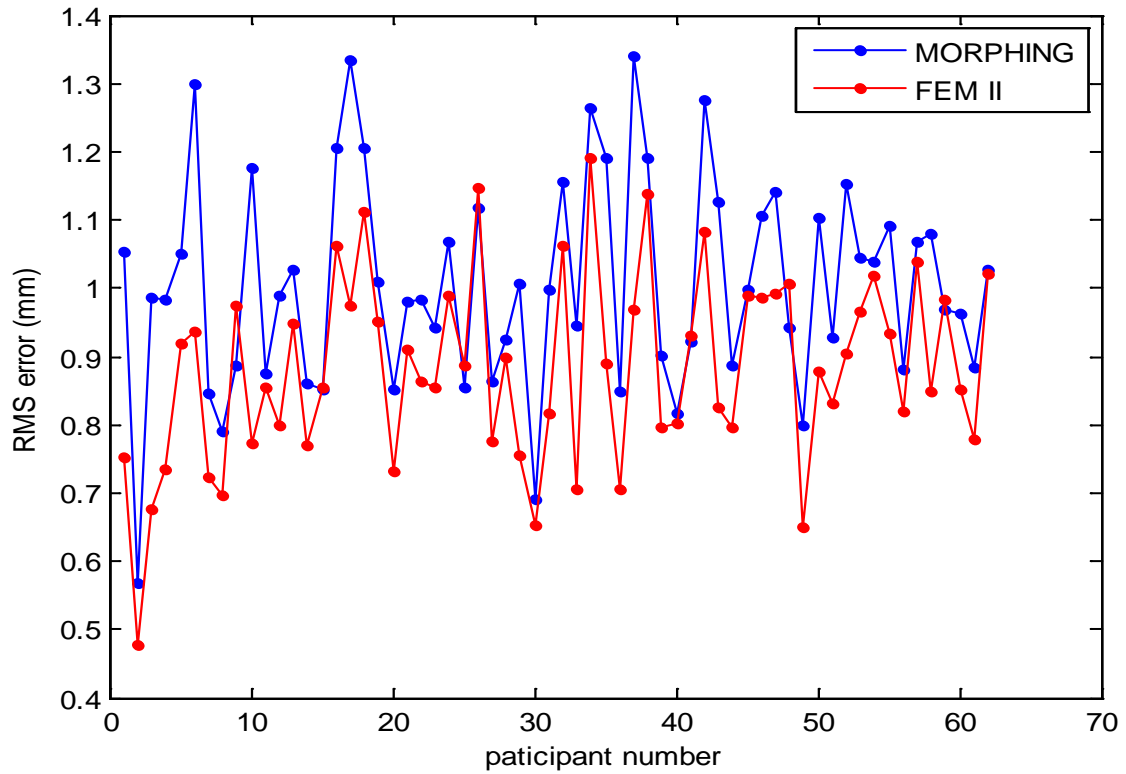
We use Z_{buffer_D}' to denote the truncated difference map.

2. Compute the RMS error. The RMS error is computed by averaging over the whole difference map:

$$E_{\text{RMS}} = \frac{\sqrt{\sum_{i,j=1}^{301} (\text{Zbuffer}'_{ij})^2}}{301}.$$

From the formula we see that when we compute the RMS error, we average over both the face area and the background where the $\text{Zbuffer}'_{ij} = 0$. This results in the effect that relatively larger faces will also have larger RMS. Thus, the comparison between the RMS values computed with different faces does not make much sense, while the comparison of RMS of different methods applied on the same face is still valid. Moreover, the average RMS error over the 62 scan pairs is a measure of the accuracy of specific method.

Below is an example chart which gives the individual RMS errors for the morphing method and FEM II. Note that the reason that we take MOPHING and FEM II is that FEM II is representative among the FEM-based method while MOPHING is a computer-vision based (non-physical) method.



After we have computed the individual RMS error for each method, to get an indication on the accuracy of each method itself, we only need to take the RMS average over the individual results.

$$E_{\text{RMS(average)}} = \sqrt{\frac{E_{\text{RMS}}}{(N_{\text{scans}})} \sum_{i=1}^{N_{\text{scans}}} (E_{\text{RMS}}(i))^2}.$$

The RMS error of each method is given in

Table 3 RMS errors

	FEMI	FEMII	FEMIII	FEMIV	MORPHING
RMS error(mm)	0.9344	0.8817	0.8882	0.8829	1.0060

5.2. Benchmarking II: Average error map

The RMS error can only provide us an averaged indication (one number) on the accuracy of each method. A more sophisticated statistics method should be applied to gain more insights in the systematic errors produced by different methods. In this section, we are going to calculate error expectation and standard deviation maps to evaluate the local performance of each method. Observe that the facial features corresponding to different people do not have identical XY coordinates, therefore the meaningful averaging of the error maps requires their morphing to the same reference. The computation is carried out as follows:

1. Create a 3D morphing mesh from the average open mouth scan landmark positions. Morph the 62 simulated open mouth and ground truth depth maps to this 3D Mesh, using the morphing technique described in section 4.2. Render the depth map $Z_{\text{buffer}_{\text{MGT}}}$ corresponding to the morphed ground truth model, and depth map $Z_{\text{buffer}_{\text{MS}}}$ corresponding to the morphed simulated model. Then compute the morphed difference map, as:

$$Z_{\text{buffer}_{\text{MD}}} = Z_{\text{buffer}_{\text{MGT}}} - Z_{\text{buffer}_{\text{MS}}}.$$

This way, we get the difference map as seen on the reference “averaged face”. Then the average error and standard deviation maps are computed in the following steps:

- Choose the background threshold value. The role of this value is to filter out meaningless values in the peripheral facial values of morphed difference map, just like we explained in section 5.1, where we assume that the absolute value of such errors should be less than 10. We use $Z_{\text{buffer}_{\text{MD}}}'$ to denote the truncated morphed difference map.
- Add up all the entries in the morphed difference maps in three ways: element-wise; element-wise squared; the number of times that each entry is above the background threshold value in the map. Thus, three new matrices are created:

$$(A_{\text{sum}})_{ij} = \sum_{\# \text{scans}} (Z_{\text{buffer}_{\text{MD}}}')_{ij}, \quad \text{for } i, j = 1, \dots, 301,$$

$$(A_{\text{sumsquare}})_{ij} = \sum_{\# \text{scans}} (Z_{\text{buffer}_{\text{MD}}}')^2_{ij}, \quad \text{for } i, j = 1, \dots, 301,$$

$$(A_{\text{count}})_{ij} = \sum_{\# \text{scans}} 1_{\{|(Z_{\text{buffer}_{\text{MD}}}')_{ij}| < 10\}}, \quad \text{for } i, j = 1, \dots, 301.$$

- Compute the average of the difference maps. Since every scan holds equal weight, the averaged error map E should just be the algebraic average.

$$E_{ij} = \frac{(A_{sum})_{ij}}{N_{scans}} \quad \text{for } \{(i,j) | (i,j) \text{ satisfies } (A_{count})_{ij} \geq 0.9 * N_{scans}\} \text{ else } E_{ij} = 0.$$

This way we only take into account the values which are “significantly” represented in all scans.

- Compute the sample standard deviation of the difference maps. According to the definition, the standard deviation σ is calculated as:

$$\sigma = \sqrt[2]{E(X^2) - E(X)^2},$$

where X denotes the random variable and $E(X)$ denotes the expectation of the random variable. The estimation of $E(X^2)$ (let us denote the matrix as E_{ij}^*) is straightforward:

$$E_{ij}^* = \frac{(A_{sumsquare})_{ij}}{N_{scans}}, \quad \text{for } \{(i,j) | (i,j) \text{ satisfies } (A_{count})_{ij} \geq 0.9 * N_{scans}\} \text{ else } E_{ij}^* = 0.$$

The **standard deviation map** is calculated as:

$$\sigma_{ij}^* = \sqrt[2]{E_{ij}^* - E_{ij}^2}, \quad \text{for } i, j = 1, \dots, 301.$$

Below, we present the average error map for each method. Please note that there are two colour scales used to present the result. We use the scale $(-2mm \text{ to } 2mm)$ for method FEM II, FEM III and FEM IV, and $(-5mm \text{ to } 5mm)$ for FEM I and MORPHING. The reason for the usage of different scale is that we want to highlight the error distribution for FEM II III and IV. If we use the same scale for all 5 methods, we would not be able to see the difference between FEM II III and IV.

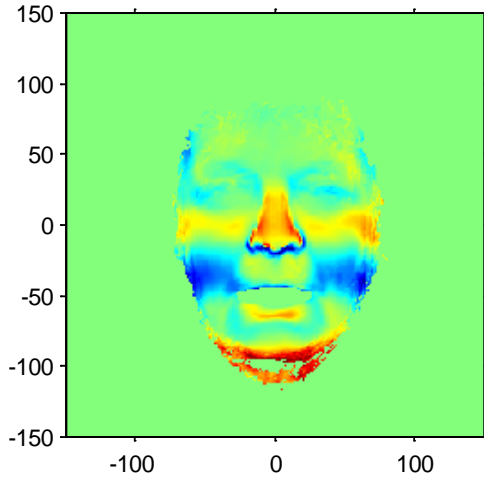


Figure 5.1 Average error map FEM III

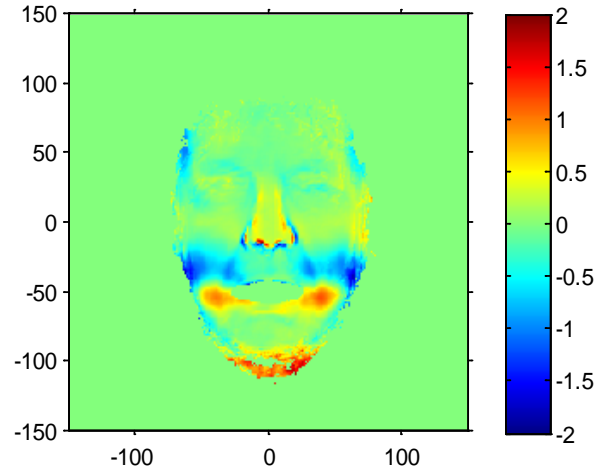


Figure 5.2 Average error map FEM II

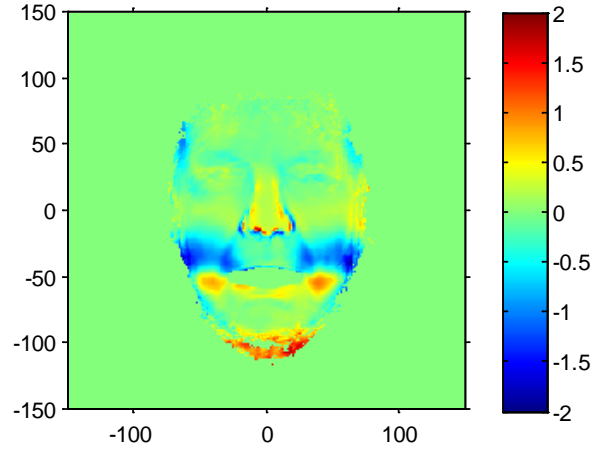


Figure 5.3 Average error map FEM VI

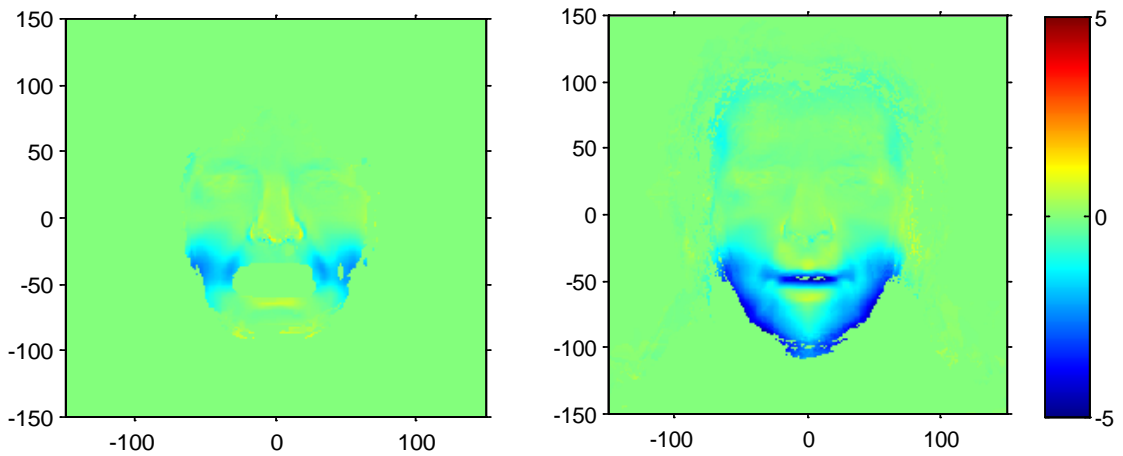


Figure 5.4 Average error map FEM I

Figure 5.5 Average error map
MORPHING

As pointed out in section 4.4, we can use the average error map for systematic correction of the simulated results in order to compensate for the systematic errors. The systematic error correction is the post-processing step where after synthesis of the open mouth scan, we add the morphed average error map to the resulting depth map.

Note that the average error map is a random array which can deviate from the expected systematic error. Therefore it is importance to check whether the systematic error correction introduce additional error or not. Thus one would like to have the variance of error maps after the correction less or equal to that without the correction. Let the systematic error X_{ij} have expectation μ_{ij} and the standard deviation σ_{ij} for $i, j = 1, \dots, 301$. Then, assuming that the individual maps X_{ij} are independent random variables, the average error map E_{ij} has expectation μ_{ij} and the standard deviation $\frac{\sigma_{ij}}{\sqrt{62}}$, where 62 is the number of individual error maps in the average. If no error correction is performed, i.e. if one assume that $\mu_{ij} = 0$, then the actual quadratic error of the measurement is equal to the second moment of X_{ij} which is equal to $\mu_{ij}^2 + \sigma_{ij}^2$. If the error correction is performed X_{ij} is reduced with E_{ij} then the second moment of $X_{ij} - E_{ij}$ is about $\frac{\sigma_{ij}^2}{62} + \sigma_{ij}^2$. Thus the correction makes sense when

$$\mu_{ij}^2 + \sigma_{ij}^2 \geq \frac{\sigma_{ij}^2}{62} + \sigma_{ij}^2,$$

which is equivalent to

$$|\mu_{ij}| \geq \frac{\sigma_{ij}}{\sqrt{62}}$$

Since the true values of the expectation and the standard deviation are not available, we replace them with the estimates which give us the practical criterion for the application of the error correction:

$$|E_{ij}| \geq \frac{1}{\sqrt{62}} \sigma_{ij}^* \approx 0.127 \cdot \sigma_{ij}^*.$$

After the systematic error correction, the random deviation from the ground truth can be purely described in terms of the standard deviation map. In this view, the standard deviation map can be used for the comparison of the competing methods. The methods with the low standard deviation are preferable.

The standard deviation maps for each method below show that the standard deviation is in general lower than the absolute value of the error map and therefore the systematic error correction does make sense.

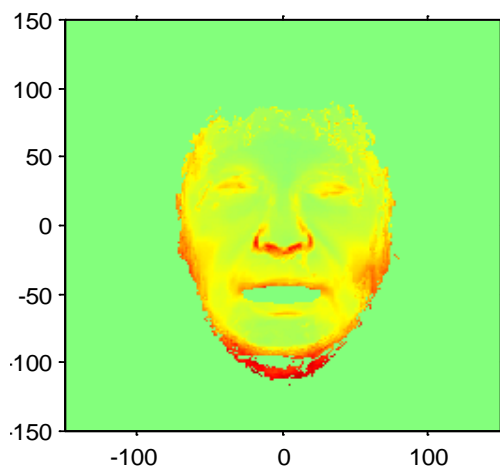


Figure 5.6 Standard deviation map
FEM III

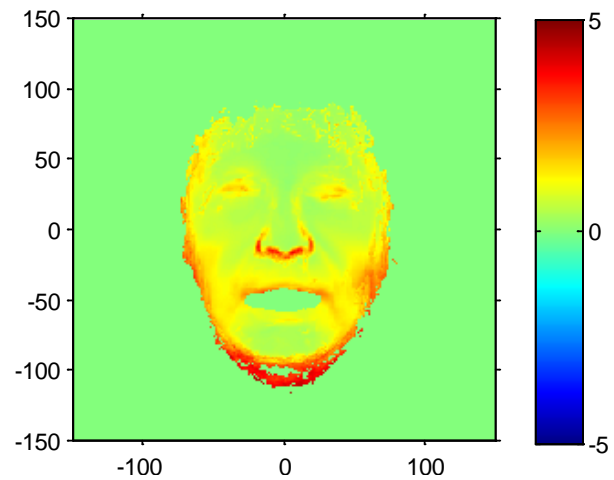


Figure 5.7 Standard deviation map FEM II

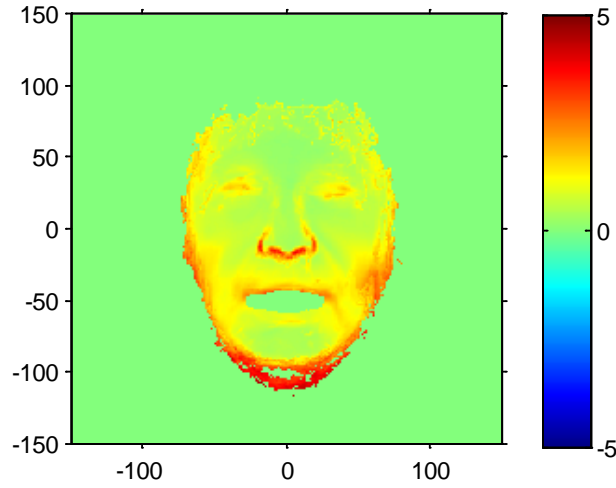


Figure 5.8 Standard deviation FEM IV

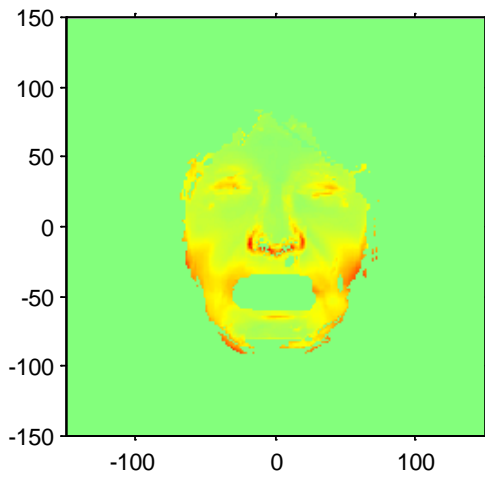


Figure 5.9 Standard deviation FEM I

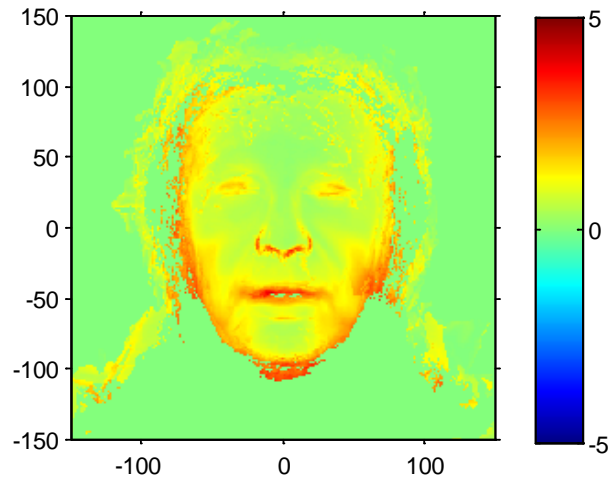


Figure 5.10 Standard deviation MORPHING

As we can see, the difference of the standard deviation map between different methods is not really visible. To compare the standard deviation maps between different methods, we use FEM II as reference. To show the relative performance of different methods with respect to FEM II we calculate the **percentage map** by subtracting the other four methods from FEM II, dividing the result with FEM II and multiplying with 100% :

$$P_{ij} = 100 * \frac{(\sigma_{FEMII} - \sigma_{method})_{ij}}{(\sigma_{FEMII})_{ij}}, \quad \text{for } i, j = 1, \dots, 301.$$

Below are the percentage maps:

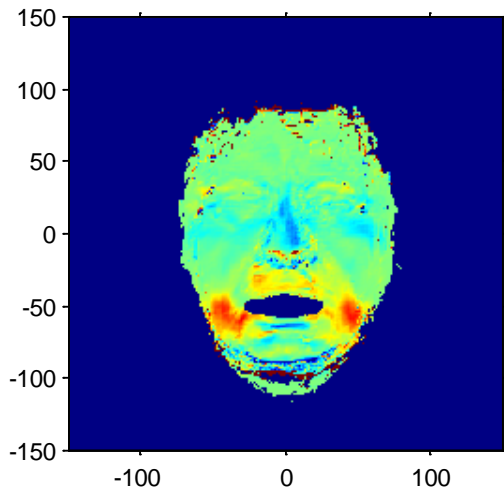


Figure 5.11. FEM II - FEM III

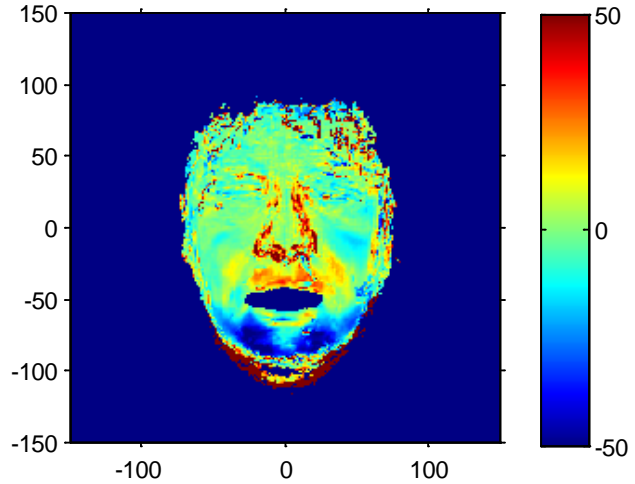


Figure 5.12. FEM II - MORPHING

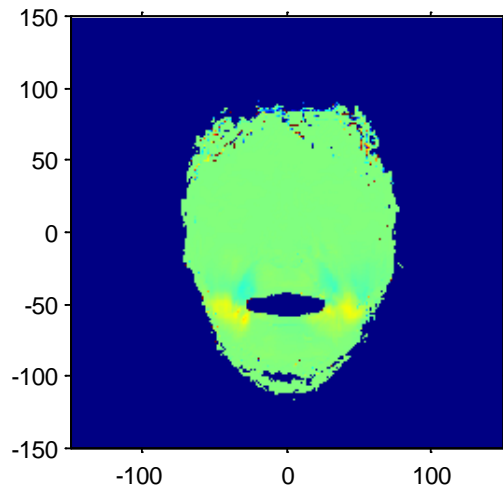


Figure 5.13. FEM II - IV

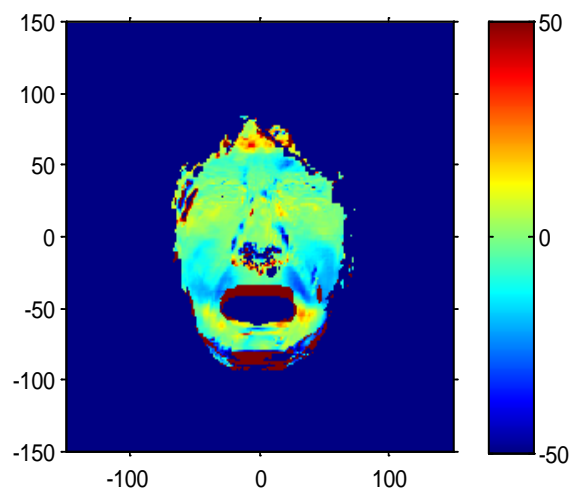


Figure 5.14. FEM II - FEM I

From the red colour on the maps indicates where locally FEM II has worse performance, and the blue areas indicate where FEMII is better.

5.3. Observations and discussions

5.3.1. Observations

From the results presented in the previous sections, the following observations can be drawn.

Observation 1: FEM-based methods are generally more accurate than the morphing method.

This fact can be observed from the RMS error. For the morphing method, the RMS error is as high as 1.006 whereas for FEM-based methods, it is only around 0.8850. Thus via RMS error estimation, FEM-based method is on average 0.1 mm more accurate than the morphing method.

The average error map also supports the observation. It can be observed from Figure 5.5 that for the morphing method, the error is mostly between 0 and 5mm, but for the FEM-based methods II, III and IV, the maximum absolute error is seldom over 2mm and mostly around 1mm. Even for FEM I, the advantage over the morphing method is visible from Figure 5.5 and Figure 5.4.

Observation 2: FEM-based methods have different error distribution than the morphing method. From the average error maps (Figure 5.11), we can clearly see that for FEM-based methods, the errors appear in a regular way namely mostly at cheeks and around the mouth, although there are minor difference between the methods. However, for the morphing based method the error mostly appear at the lower face and it appears more scattered than the FEM-based methods. This fact is confirmed by Figure 5.12, where we observed that FEM II has around 50% less standard deviation in error than morphing method in the chin area.

Observation 3: Within the FEM-based methods, FEM I is the least accurate method.

A check on both RMS error estimation and average error map like we did in observation 1 yields the result.

Observation 4: FEM II and FEM IV are very similar in terms of error distribution.

Indeed, in RMS error indication, the difference between the two methods is only 0.0012mm . Moreover, we can barely see the difference in the average error maps for the two methods. The percentage map (Figure 5.13) also confirms the statement by showing only around 10% difference in standard deviation map at cheek area.

Observation 5: In terms of the standard deviation of the errors, FEM II, III and IV outperform the morphing method in the chin area.

This observation is derived from the percentage map. In the morphing method's percentage map blue colour dominates in the chin area (which means higher standard deviation values than FEM II), while FEM IV is very similar to FEM II and FEM III is better than FEM II in the cheek area (as indicated by red colour in the percentage map in red (which means less standard deviation than FEM II)).

Observation 6: FEM III has the best average performance in standard deviation.

This observation is drawn from the percentage maps. Indeed, FEM III is better than FEM II on the mouth sides, and therefore is also better than FEM IV which is very close to FEM II, FEM I is worse than FEM II and therefore FEM I is also worse than FEM III. One can find that the advantage of FEM III over the Morphing in the chin area is about 40% after the correction of systematic errors. However, without the compensation of the systematic errors FEM III performs the worst among FEM II, III and IV in terms of the average error where FEM III has 0.0065mm more error than FEM II and 0.0053mm more than FEM IV. In the average error map, we also see that only FEM III has visible error at the upper face (between 0.5mm and 1mm) which leaves room for improvements and optimizations.

5.3.2. Discussions

Based on above observations, we would like to discuss the following aspects:

1. *Physical-based model versus computer-vision method*

From the observation 1, we can conclude that the physical-based model is generally more accurate than the computer-vision based method, though the former costs more computational power. We observe that, in absolute values, the average advantage of FEM III over the Morphing is about 0.5mm after correction of the systematic errors. At the same time, the FEM method is considerably more accurate without statistics based systematic correction of the simulation errors. Therefore The FEM based methods are preferable for simulations which do not have ground truth data, like, for example, the side shifts of the jaw.

2. *Shell element versus solid element*

Shell element is a structural element. It is also a very common choice of engineers. But

observation 3 tells us that it performs worse compared to 3D continuum solid element in our models. This is not a surprise. Though the doubly curved shells we used can represent objects in three dimensions virtually, they are surface based element in essence. Through defining the nodal coordinates and the thickness the user gets the interpolated surface. Especially, in the shell-element based model, we retain the computed facial surface using the original soft tissue thickness for the neutral face (since this is no other ways to regain the simulated face surface), we have partially lost the quasi-incompressible property of the facial tissue. Since the quasi-incompressible property will lead to the loss of thickness in soft tissue when it gets stretched during the mouth opening process. This could also be a possible reason for shell elements to lose their accuracy.

3. St.Venant-Kirchhoff model versus Mooney model versus Gent model

From observations 4,5 and 6, we can conclude that the St.Venant-Kirchhoff model (FEM II) does not underperform other two material models, although Mooney (FEMIII) and Gent (FEM IV) models have a non-constant elasticity tensor. This can be explained by the fact that during the mouth opening process the skin elements experience large displacement but not sufficiently large strain. In the small strain regime, the skin has a linear relationship between strain and stress, as shown in Figure 3.6. Thus the St.Venant-Kirchhoff model does not show an obvious disadvantage compared with the other two models.

We also know from Observation 4 that the Gent model behaves quite similar to the St.Venant-Kirchhoff model. This is probably caused by the similarity of the St.Venant-Kirchhoff model and the Gent model in their strain-stress relationship in the small strain regime. In the section “**Material models and uniaxial tension test**”, we can observe the linear relationship between strain and stress in the small strain area for the Gent model.

Observation 6 tells us that the Mooney material model performs best in terms of the standard deviation. Unfortunately, we cannot provide a satisfactory explanation on this yet at the moment.

6. Conclusion and future work

We have developed five methods in total which can simulate open mouth facial scans from the closed mouth scans. Except from the last morphing method, the other four methods are based on 3D finite element tissue models which allow a reasonable prediction of tissue deformations according to the open mouth process. The models are built based on individual facial surface scans and generic soft tissue thickness data (which can be taken from the Philips MRI data study or the open literature study). Using this information, we created four models in Lagrangian formulation. The models incorporate geometric nonlinearity and mechanical nonlinearity of the biological tissues. We verified our simulation results with the actual open mouth scans and calculated the averaged error and standard deviation maps for each method. For physically-based models, the largest error typically occurs at the cheek area and is within $[-2mm, 2mm]$.

To develop a more powerful computational tool for mouth opening, realistic soft tissue models are desirable. To improve the performance of the existing soft tissue models, we recommend to look into the following aspects:

1. Multi-layer soft tissue model

Multi-layer models can be developed to represent different biological soft tissue layers (e.g. epidermis layer). A substantial amount of work has been done in this direction [21]. Multi-layer models, theoretically speaking, with proper assignment of the parameters, will model the non-homogeneity better than double layer models.

2. Muscle forces activation

Muscle effect will also improve the result. Like pointed out in section **Facial soft tissue anatomy**, open mouth process involves activation of muscle forces (although it is not predominant). Thus, a proper muscle force activation model can make the mouth open process more realistic. Besides, adding the muscle layer will also increase the realism structurally. A lot of effort has been put into studying muscle force models. Related literature is [11, 22].

3. Anisotropic property

In our models, we have assumed the isotropic property, but the real soft tissue layers are anisotropic. In Marc Mentat, the user can specify the fourth-order elasticity tensor for an anisotropic material. For example in linear case, user can specify the fourth order tensor C_{ijrs} , which gives the strain-stress relation $\sigma_{ij} = C_{ijrs}\epsilon_{rs}$.

4. Material postulation

In our project, we used Moony and Gent, two hyperelastic material models to simulate the complex biological object - human facial soft tissue. As we mentioned in section 3.2.2, a nonlinear material constitutive relation is determined by the strain density function. Hence, for a material as complex as human skin, to postulate a strain energy function based on experimental data is a way to set up a mathematical model. For more details, see [29].

5. Nonlinear springs

Due to superficial fascia, there exists “sliding effect” when people open their mouth. In our model, we use springs to simulate the “sliding effect”. However, the sliding effect is more than a linear relationship between displacements and forces. In Marc Mentat, it is possible to define nonlinear springs using a table to model nonlinear boundary conditions. That will give us more freedom to model the “sliding effect”.

6. Model for superficial fascia

In our model we used springs to model the superficial fascia in our effort of replicating the “sliding effect”. More advanced models exist to simulate the fibrous connective tissues [43].

7. Gravity effect

We have ignored all the body forces in the models. But in reality all humans experience gravity. Especially for people who have a lot of fat in their face, when they lie down, probably the gravity will play an important role in the geometry of the face. Thus if we add body force (gravity) into the model, we may achieve a better result.

8. Higher-order finite element

In the project, we have employed linearly-interpolated elements. Higher order elements will increase the accuracy, but at the same time the computational effort will also increase. A more accurate quadrature rule can also be utilized in combination of the higher-order element.

9. Adaptive meshing

To reduce the computational effort, we can use various meshing methods to reduce the number of nodes. In other words, for the region for which we do not expect too much change (like the upper head) we can put less nodes there.

10. Participants classification

In section 3.1.1, we have pointed out that the properties of skin differ with age, gender, race etc. Taking above mentioned information of a participant into account when we design our material model would be helpful especially when aiming at a personalized mask sizing advice.

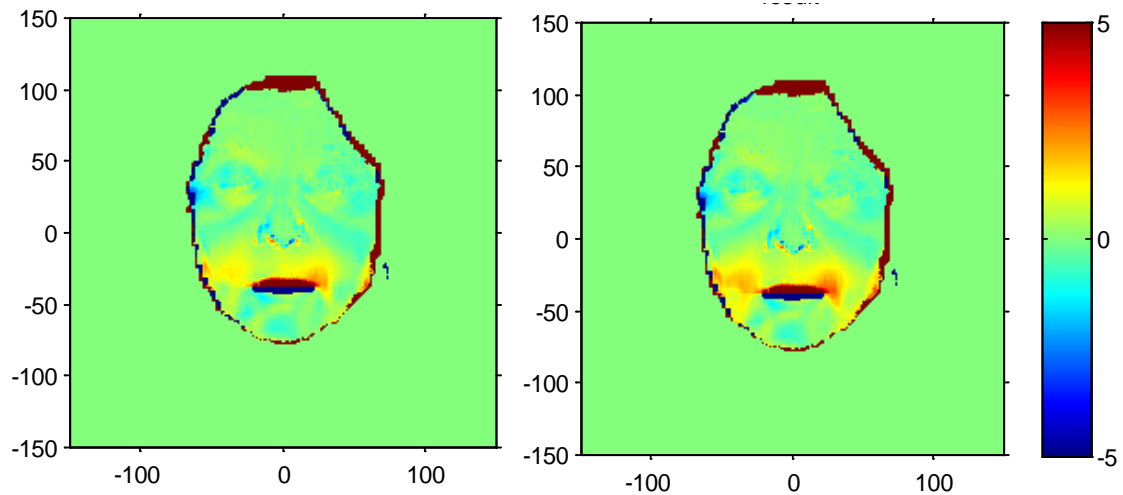
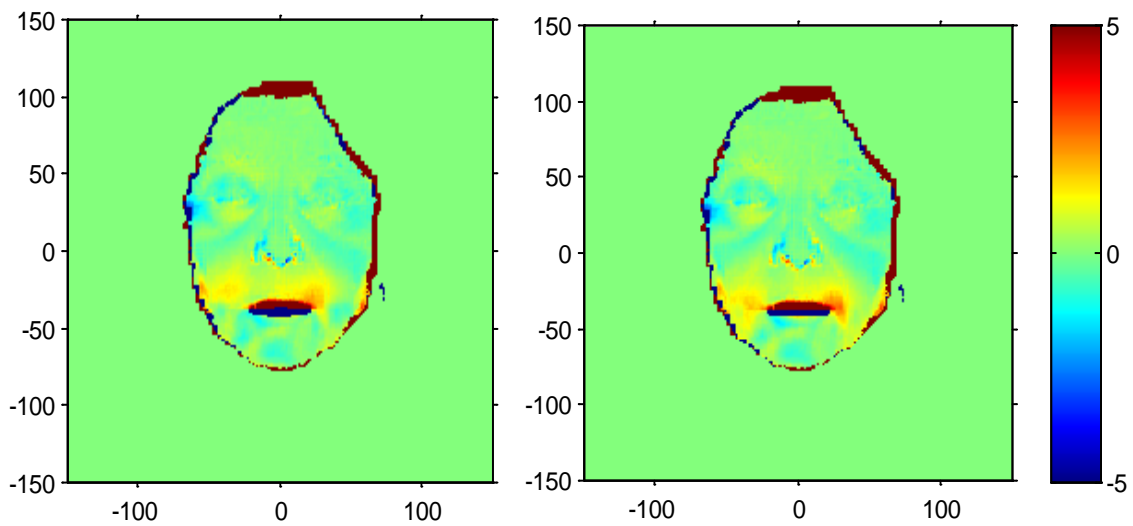
11. Parameter optimization

Given a model configuration, the parameters determine the error produced by the model. In the project, we used parameters values found in the literature. Different sets of parameters will give different amplitude and distribution of the errors over the error maps. Thus, parameter tuning plays an important role.

Below we consider the St.Venant-Kirchhoff model (FEM II) as an example to illustrate the effects of parameters. Note that there are three parameters involved in the model: Young's modulus (E), Poisson ratio (ν) and spring stiffness (k). Table 4 below shows the parameter settings for four different simulations. The resulting error maps are shown in Figure 6.1 and Figure 6.2.

Table 4 Parameter tuning

	Young's modulus (E)	Poisson ration (ν)	spring stiffness (k)	RMS error
Case 1	0.5	0.42	1	79.59890
Case 2	0.5	0.22	1	94.52781
Case 3	0.5	0.42	2	78.91196
Case 4	2.5	0.42	1	84.13046

*Figure 6.1 Case 1&2**Figure 6.2 Case 3&4*

From this example, we get an idea about how parameters affect our result. Thus, a significant improvement for the model might be gained from the optimization of the parameters. With the current model configuration, one can try to find the optimal set of parameters which will minimize the specified error measure.

7. Acknowledgement

I would like to thank all the people who helped and supported me to finish this thesis. These people include: Dmitry Znamenskiy, supervisor at Philips research, who basically guided me through the whole project; Ruud Voncken, supervisor from Philips, who provided generous help with all sorts of Marc-related problems; Barry Koren, supervisor from TU/e, who gave me a lot of useful comments on writing the thesis.

I also would like to show my gratefulness to my friends who have given valuable support both mentally and technically when I met difficulties in this project.

At last, a whole-hearted thank you to my parents. Their support made my master study in Eindhoven possible and their selfless care and love to me during this project gave me strengths to challenge all the difficulties in this project and not to give up.

A. Appendices

A.1 Model definition file

The header of the Marc model input file looks like:

```

title                                standard
extended
sizing                               0  5823  5992  0
alloc                                25
elements                             75
version                              11
processor                             1    1    1    0
all points
shell sect                           5    0    1
end
solver
      8    0    0    0    0    0    0    0    0    0    0    0
optimize                             11

```

where 5823 is the total number of elements and 5992 is the total number of nodes.

The header is followed by describing how the nodes are connected into the elements:

```

connectivity
      0    0    1
      1    7  2166  2267  2268  2167  12382  12483  12484  12383
      2    7  2167  2268  2269  2168  12383  12484  12485  12384
      3    7  2168  2269  2270  2169  12384  12485  12486  12385

```

.....

where the first number in the row is the element number, 7 corresponds to “Hexahedral element” type, and the last 8 numbers are the indices of the nodes. Since the skull is modelled with the shell elements, the corresponding connectivity part looks like:

```

      2392    75  20433  20447  20448  20434
      2393    75  20434  20448  20449  20435
      2394    75  20435  20449  20450  20436
.....
      4782    75  22918  22935  22936  22919

```

where 75 in the second row corresponds to the “Shell element” type consisting of only 4 nodes.

The connectivity part is followed by the table of node coordinates:

```

coordinates
      3  7580    0    1
      2166 -1.8000000000E+01 -8.7000000000E+01 -6.2510218835E+01
      2167 -1.5000000000E+01 -8.7000000000E+01 -5.5302429785E+01
      2168 -1.2000000000E+01 -8.7000000000E+01 -5.2346172906E+01
.....

```

where the first column contains the node indices, and the remaining contain the XYZ node coordinates in the Cartesian coordinates system.

Next, the skin node set is defined

```

define  node          set          skin

```

The declaration is followed by the list on node indices where 'C' character indicates that there are more indices on the next line

```
2166 2167 2168 2169 2170 C
2171 2172 2173 2174 2175 C
2176 2177 2265 2266 2267 C
```

.....

This is repeated for the upper and lower skull parts:

```
define node set upperskull(shell)
20872 20873 20874 20875 20876 C
20877 20878 20879 20880 20881 C
20882 20883 20884 20885 20886 C
```

....

```
define node set lowerskull(shell)
20433 20434 20435 20436 20437 C
20438 20439 20440 20441 20442 C
20443 20444 20445 20446 20447 C
```

.....

Next lines define the material properties of the elements

isotropic

```
1elastic 10 0 0 0material1
```

"Isotropic" option allows the user to assume isotropic property of the material.

```
5.000000000000E-01 4.200000000000E-01 0.000000000000+0 0.000000000000+0 0.000000000000+0
0.000000000000+0 0.000000000000+0 0.000000000000+0 0.000000000000+0
1 to 4782
```

The first two numbers after the declaration specify the Young's modulus and Poisson's ratio, respectively. Then geometry of the skin is defined:

geometry

```
0.000000000000+0 0.000000000000+0 0.000000000000+0 0.000000000000+0 0.000000000000+0
0.000000000000+0 0.000000000000+0 0.000000000000+0 0.000000000000+0
1 to 2391
```

followed by the geometry of the skull:

```
2.000000000000E-01 0.000000000000+0 0.000000000000+0 0.000000000000+0 0.000000000000+0
0.000000000000+0 0.000000000000+0 0.000000000000+0 0.000000000000+0
2392 to 4782
```

where the first number after the declaration specifies the (uniform) thickness of the elements indicated in the second part (element from 2392 to 4782).

Then the springs are defined:

springs

```
12382 1 20433 11.000000000000E+00 0.000000000000+0 0.000000000000+0
0.000000000000+0 0.000000000000+0 1 0
0 0 0 0 1 1 0 0
```

...

```
12383 1 20434 11.000000000000E+00 0.000000000000+0 0.000000000000+0
0.000000000000+0 0.000000000000+0 2 0
0 0 0 0 1 1 0 0
```

In the "springs" option, the first column gives the node number of the first end of the spring and the third place for the second end. The second and the fourth positions specify the degree of the freedom at the nodes that indicated before it. The next position is the stiffness of the springs (for

linear case). At the tenth position, the spring ID is flagged. The end of a Marc input file looks like:

```
no print
end option
```

A.2 Linux subroutine

```
% use this script to run the simulation in marc
% first check the file is ready or not
k=1;
while k<2
while exist('mouth_opening_face_input.dat','file')~=2
    pause(0.5);
end
f=dir('mouth_opening_face_input.dat');
while f.bytes<=220000
    pause(0.5);
    f=dir('mouth_opening_face_input.dat');
end

% run simulation in mentat
!mentat make_mouth_opening_model_V2.proc

pause(30);
end
```

A.3 Marc procedure file

```
| Created by Marc Mentat 2012.1.0 (64bit)
|*prog_option compatibility:prog_version:ment2012
|-----
| Procedure file for creating mouth opening FEM model
| Input:
|   - Parameter file "mouth_opening_model_parameters.inproc"
|   - Database mouth_opening_face_input.mfd containing:
|       mesh of faces
|       mesh of skull parts
|       springs
|-----
|
|-----
| Settings
|-----
|
|*reset
|*new_model yes
|
|*colormap 2
|*set_links off
|*set_nodes off
```

```

*elements_solid
*set_aplys off
*reset_view
*rot_model_cspace_y_rev
*rot_model_cspace_y_rev
*rot_model_cspace_y_rev
*rot_model_cspace_y_rev
*rot_model_cspace_x_for
*rot_model_cspace_x_for
|
| -----
| Read parameter file
| -----
|
*exec_procedure mouth_opening_parameters.inproc
|
| -----
| Add rotation points for lower skull
| -----
|
*add_nodes
0 0 0
|
*select_clear
*select_nodes
1 #
*store_nodes rotation_node
all_selected
*select_reset
|
| -----
| Merge database with face description
| -----
|
*set_merge_renumber off
|
|*merge_model mouth_opening_face_input.mfd
*import marc_read mouth_opening_face_input.dat
*fill_view
|
| -----
| Save database
| -----
|
*save_as_model mouth_opening_simulation.mud yes
|
| -----
| Defining element sets
| -----

```

```

|
*select_elements_nodes skin
*store_elements skin_elements all_selected
*select_clear_elements
|
*select_elements_nodes lowerskull(shell)
*store_elements lower_skull_elements all_selected
*select_clear_elements
|
*select_elements_nodes upperskull(shell)
*store_elements upper_skull_elements all_selected
*select_clear_elements
|
| -----
| Defining geometries
| -----
|
*identify_sets *regen
*geometry_name skin_thickness
*add_geometry_elements
*identify_geometries *regen
|
*new_geometry *geometry_type mech_three_shell
*geometry_name skull_thickness
*geometry_param thick
1
*add_geometry_elements lower_skull_elements upper_skull_elements
|
| -----
| Defining boundary conditions
| -----
|
| Clamp upper skull
|
*new_apply *apply_type fixed_displacement
*apply_name clamp_upper_skull
*apply_dof x *apply_dof_value x
*apply_dof y *apply_dof_value y
*apply_dof z *apply_dof_value z
*apply_dof rx *apply_dof_value rx
*apply_dof ry *apply_dof_value ry
*apply_dof rz *apply_dof_value rz
*add_apply_nodes upperskull(shell)
|
| Define linear table
|
*new_md_table 1 1
*set_md_table_type 1
time
*table_add

```



```

0 0
1 1


```

```
*loadcase_name Mouth_opening
*loadcase_value maxrec 100
*loadcase_option initstress:tensile
|
|-----
| Defining job
|-----
|
*update_job
*add_job_loadcases Mouth_opening
*add_job_applys clamp_upper_skull
*add_job_applys Rotation_point_activation
*job_option strain:large
*job_option post:binary
*job_option impd:on *job_option elevar:on
*job_usersub_file subrou_V1.f
*add_post_var ecauchy
*add_post_var eel_strain
*update_job
|
|-----
|-----
|
*save_model
|
*job_submit_reset
*submit_job 1 *monitor_job
|
*quit yes
```

References

- [1] K.Komatsu. Human Skin Model Capable of Natural Shape Variation. The visual computer, Vol.3. p. 265-271,1988.
- [2] Lee Y. Terzopoulos D. Waters K. Realistic Modelling for Facial Animation. SIGGRAPH '95 Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, p. 55 – 62,1995.
- [3] Girod B. Keeve E., Girod S. Craniofacial Surgery Simulation. In proceedings of the 4th International Conference on Visualization in Biomechanical Computing VBC'96, p. 541-546, Hamburg, Germany, September 1996.
- [4] Keeve E. Girod S. Kikinis R. Girod B., Deformable Modelling of Facial Tissue for Craniofacial Surgery Simulation. Computer Aided Surgery. 1998, Vol. 3, No. 5 , p. 228-238.
- [5] Bourguignon D. Cani M. Controlling Anisotropy in Mass-Spring Systems. Computer Animation and Simulation 2000 Eurographics 2000, p. 113-123.
- [6] Delingette H. Towards Realistic Soft Tissue Modeling in Medical Simulation. Proceedings of the IEEE. Volume 86, Issue 3, p. 512- 523.
- [7] Wayne F. Larrabee, JR. MD. A Finite Element Model of Skin Deformation. Laryngoscope 96, April 1986.
- [8] X.Q.Deng. A Finite Element Analysis of Surgery of the Human Facial Tissue. Ph.D thesis, Columbia University, New York, 1988.
- [9] Keeve E. Girod S. Pfeile P. Girod B. Anatomy-Based Facial Tissue Modelling Using the Finite Element Method. Proceedings of Visualization ' 96 Oct. 27 1996-Nov.1 1996 p. 21- 28.
- [10] Koch R. Gross. M Carls F. Simulating Facial Surgery Using Finite Element Models. SIGGRAPH '96 Proceedings of the 23rd annual conference on Computer graphics and interactive techniques .p.421-428.
- [11] Koch M. Gross M. Bosshard A. Emotion Editing Using Finite Elements. Computer Graphics Forum, Volume 17, Issue 3, p. 295–302, September 1998.
- [12] Koch R. Gross M. Zimmermann A. Sailer H. A Framework for Facial Surgery Simulation. SCCG '02 Proceedings of the 18th spring conference on Computer graphics, p. 33 – 42.
- [13] Roth S. Gross M. Turello S. Carls F. A Bernstein-Bézier Based Approach to Soft Tissue Simulation. Eurographics '98. Volume 17, Number 3 ,1998.
- [14] Roth SHM. Bernstein-Bezier Representation for Facial Surgery Simulation. PhD Thesis, Swiss Federal Institute of Technology, 2002.
- [15] Nebel J. Soft Tissue Modelling from 3D Scanned Data. Deformable Avatars, p. 85-97. Springer US, 2001.
- [16] Cotin S. Delingette H. Ayche N. A Hybrid Model Allowing Real-time Cutting, Deformation and Force-Feedback for Surgery Training and Simulation. Visual Computer, Vol 16, No.8 p. 437-453, 2000.
- [17] Picinbono G. Delingette H. Ayache N. Non-linear and Anisotropic Elastic Soft Tissue Models for Medical Simulation. Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference 2001. Volume 2. p. 1370-1375.
- [18] Gladilin E. Zachow S. Deuflhard P. Hege H. On Constitutive Modeling of Soft Tissue for the Long-Term Prediction of Cranio-Maxillofacial Surgery Outcome. International Congress Series 1256 (2003). p. 343-348.
- [19] Chabanas M. Luboz V. Payan Y. Patient Specific Finite Element Model of the Face Soft Tissues for Computer-Assisted Maxillofacial Surgery. Medical Image Analysis. 2003 June. Volume 7 Issue 2, p.131-51.
- [20] Wang S. Yang J. Simulating Cranio-Maxillofacial Surgery Based on Mixed-Element Biomechanical Modelling. Computer Methods in Biomechanics and Biomedical Engineering. Volume 13, No. 3, June 2012. p. 419-429.
- [21] Hung A. Mithraratne K. Sagar M. Hunter P. Multilayer Soft Tissue Continuum Model: Towards Realistic Simulation of Facial Expressions. World Academy of Science, Engineering & Technology;Jun2009, Vol. 54, p.134.
- [22] Beldie L. Walker B. Lu Y. Richmond S. Middleton J. Finite Element Modeling of Maxillofacial Surgery and Facial Expressions-a Preliminary Study. The international journal of medical robotics

- and computer assisted surgery. Vol. 6. p. 422-430, 2010.
- [23] James DL, Pai DK. Multi-resolution Green's Function Methods for Interactive Simulation of Large-Scale Elastostatic Objects. *ACM Trans Graph* 2003, 22(1) p.47-82.
- [24] Bo Zhu, Lixu Gu. A Hybrid Deformable Model for Real-Time Surgical Simulation. *Computerized medical imaging and graphics* 36, 2012, p. 356-365.
- [25] Jansson, J. and Vergeest, J.S.M. A Discrete Mechanics Models for Deformable Bodies. *Computer-Aided Design*. Amsterdam, 2002.
- [27] M.D. Frank, H. Netter. *Atlas of Human Anatomy*. Icon Learning System Teterboro, New Jersey, 3 edition, 2003.
- [28] Evgeny Gladilin. *Biomechanical Modelling of Soft Tissue and Facial Expressions for Craniofacial Surgery Planning*. PhD thesis. Free University of Berlin, 2002.
- [29] Yuan-Cheng Fung. *Biomechanics: Mechanical Properties of Living Tissues*. 2nd edition. Springer, Berlin, 1993.
- [30] F.M. Hendriks. *Mechanical Behaviour of Human Skin in Vivo*. Nat.Lab. Unclassified Report 2001/820. Koninklijke Philips Electronics N.V. 2001.
- [31] W. Maurel, Y. Wu, N. Magnenat Thalmann, and D. Thalmann. *Biomechanical Models for Soft Tissue Simulation*. Esprit Basic Research Series. Springer-Verlag, Berlin, 1998.
- [32] C.H. Daly. Biomechanical Properties of Dermis. *The Journal of Investigative Dermatology*, 79: p.17s-20s, 1982.
- [33] Morton E. Gurtin. *An Introduction to Continuum Mechanics*. Academic Press, London 1981.
- [34] Javier Bonet, Richard D. Wood. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press, Cambridge, 2008.
- [35] Marc Manual 2012 Volume A
- [36] Klaus-Jürgen Bathe. *Finite Element Procedures for Solids and Structures*. MIT Open Course Ware, <http://ocw.mit.edu>.
- [37] Marc Manual 2012 Volume B
- [38] Klaus-Jürgen Bathe. *Finite Element Procedures*. Prentice Hall, Englewood Cliffs, 1996.
- [39] Thomas J.R. Hughes. *The Finite Element Method : Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall, Englewood Cliffs, 1987.
- [40] Eugenio Oñate. *Structural Analysis with the Finite Element Method : Linear Statics*. Springer Netherlands, 2009.
- [41] Marc product website: <http://www.mscsoftware.com/product/marc>.
- [42] M. Wu, D. Znamensky, R. Jasinschi. *Analysis of Facial Soft Tissue Distribution with 3D MRI Data*. Technical Note, Philips.
- [43] Lanir, Y. Constitutive Equations For Fibrous Connective Tissues. *Journal of Biomechanics*, 1983, 16, p. 1-12.
- [44] Darijani, Hossein, and Reza Naghdabadi. Hyperelastic Materials Behavior Modeling Using Consistent Strain Energy Density Functions." *Acta mechanica* 213.3-4 (2010): 235-254.