Thesis submitted to obtain the title of
Doctor of Philosophy

Doctoral School of Engineering Science
Field: Computer Science

Real-time Soft Tissue Modelling on GPU for Medical Simulation

Prepared by Olivier Comas at
INRIA Lille, Shaman Team and CSIRO ICT Brisbane, AEHRC

DRAFT - Sat 7th of August 2010 at 17:15

Jury:

Reviewers:	Bernard	-	INRIA (Shaman)
	Bernard	-	INRIA (Shaman)
Advisor:	Stéphane Cotin	-	INRIA (Shaman)
President:	Bernard	-	INRIA (Shaman)
Examinators:	Bernard	-	INRIA (Shaman)
	Bernard	-	INRIA (Shaman)

Contents

Contents i

I Introduction 1

1 Medical simulation 3

1.1 General context 3

1.1.1 Simulators for everything: we started by simulating what is simple 3

1.1.2 As complexity increases, applications to medical training, patient-specific planning and per-operative guidance 3

1.2 Challenges 3

1.2.1 Photorealistic rendering 3

1.2.2 Accurate haptic feedback 3

1.2.3 Physically realist organ modelling in real-time 3

2 Background in continuum mechanics for soft-tissue modelling 5

2.1 Introduction 6

2.2 Description of motion 6

2.2.1 Lagrangian description 7

2.2.2 Eulerian description 7

2.2.3 Displacement field 8

2.3 Analysis of deformation 8

2.3.1 Deformation gradient tensor 8

2.3.2 Change of volume 9

2.3.3 Change of surface 10

2.3.4 Volumetric and isochoric components 10

2.4 Strain measures 11

2.4.1 Cauchy-Green deformation tensors 11

2.4.2 Green strain tensor 11

2.4.3 Cauchy and Euler tensor 12

2.4.4 Principal strains and invariants 13

2.4.5 Infinitesimal strain tensor 13

2.5 Stress 14

2.5.1 Cauchy stress 14

2.5.2 First Piola-Kirchhoff stress tensor 16

2.5.3 Second Piola-Kirchhoff stress tensor 16

2.5.4 Principal stresses and invariants 17

2.6 Constitutive equations 17

2.6.1 Elasticity 18

2.6.2 Linear materials: generalised Hooke's law 18
2.6.3 Orthotropic materials 19
2.6.4 Isotropic materials 20
2.6.5 Non-linear materials 20
2.6.6 Viscoelastic materials 21
2.7 Tissue characterisation 22

3 Practical approach of the Finite Element Method 23
3.1 Introduction 24
3.1.1 A numerical method 24
3.1.2 The basic ideas of FEM 24
3.2 Discretisation 25
3.2.1 Meshing process 25
3.2.2 Solution interpolation 26
3.2.3 Natural coordinates 27
3.2.4 Geometry interpolation 28
3.2.5 A particular case: isoparametric elements 29
3.3 Derivation of element equations 29
3.3.1 Strong and weak forms 29
3.3.2 Time dependence 30
3.3.3 Dynamic system of equations 30
3.3.4 Static system of equations 31
3.3.5 A few words on the matrices involved 31
3.4 Assembly of element equations 33
3.5 Solution of global problem 34
3.5.1 Explicit time integration 34
3.5.2 Implicit time integration 36
3.5.3 Static solutions 37
3.5.4 Solvers 37

II Solid organs modelling 43

4 State of art: FEM 45
4.1 Expose the problem: how to model solid organs? 45
4.2 Techniques using geometrical kind of approaches 45
4.2.1 Shape matching 45
4.3 Techniques relying on physics 45
4.3.1 Mass-spring 45
4.4 Techniques based on continuum mechanics 45
4.4.1 FEM with mesh (highlight evolution: linear, topological changes, co-rotational, non-linear, GPU) 45
4.4.2 Meshless (supposedly good for handling topological changes but other issues 45

Olivier Comas - Sat 7$^{\text{th}}$ of August 2010 at 17:15

5 Introduction of TLED 47
5.1 Description of the algorithm (neo-Hookean, explicit integration, independent elements, parallelisable on GPU) 47
5.2 Visco-elasticity and anisotropy added (MICCAI 2008; MedIA 2009) 47

6 GPU implementation of TLED 49
6.1 What is GPGPU 49
6.2 Re-formulation of the algorithm for its Cg implementation 49
6.3 CUDA implementation/optimisations (ISBMS 2008a) 49
6.4 Implementation into SOFA 49
  6.4.1 Presentation of SOFA project and architecture 49
  6.4.2 TLED in SOFA and release in open-source 49

III Hollow organs modelling 51

7 State of art: hollow structures 53
7.1 Expose the problem: models presented for solid organs are insufficient. How to model hollow organs? 53
  7.1.1 Colonoscopy simulator project: needs for colon 53
  7.1.2 Cataract surgery, stenosis: other needs for implants/blood vessels 53
7.2 Non-physic approaches (computer graphics stuff) 53
7.3 Physically accurate approches (plates/shells) 53

8 Mass-spring model for colon implemented on GPU (ISBMS 2008b) 55
8.1 Colon is big, we need a fast model 55
8.2 Limitations (difficult to link it to actual physical properties) 55

9 A co-rotational triangular shell model (ISBMS 2010) 57
9.1 Model description 57
9.2 Validation 57
9.3 Application to implant deployment simulation in cataract surgery . . 57

10 'Shell meshing' technique (MICCAI 2010) 59
10.1 State of art: reconstruction/simplification 59
10.2 Our method 59

IV Conclusion 61

11 Summary 63

12 Discussion and perspectives (Interaction solid/hollow organs) 65

A Tensors 67

B The weighted residual method **69**

References **71**

Part I

Introduction

Olivier Comas - Sat 7$^{\text{th}}$ of August 2010 at 17:15

CHAPTER 1

MEDICAL SIMULATION

1.1 General context

1.1.1 Simulators for everything: we started by simulating what is simple

1.1.2 As complexity increases, applications to medical training, patient-specific planning and per-operative guidance

1.2 Challenges

1.2.1 Photorealistic rendering

1.2.2 Accurate haptic feedback

1.2.3 Physically realist organ modelling in real-time

Chapter 2

Background in continuum mechanics for soft-tissue modelling

As seen in the previous chapter, realistic modelling of organ deformation is a challenging research field that opens the door to new clinical applications including: medical training and rehearsal systems, patient-specific planning of surgical procedures and per-operative guidance based on simulation. In all these cases the clinician needs fast updates of the deformation model to obtain a real-time display of the computed deformations. If for medical training devices the haptic feedback from touching organs merely needs to feel real, the accuracy of the information provided to the clinician in the cases of planning or per-operative guidance is crucial. Therefore, a substantial comprehension of the mechanics involved and a knowledge of the physical properties of anatomical structures are both mandatory in our quest to realistically model the deformation of organs. This chapter will start by introducing the main concepts of continuum mechanics that are fundamental to study the mechanical response of organs. It will then present mathematical models able to describe the different mechanical aspects of materials and briefly discuss the mechanical characterisation of tissues.

2.1 Introduction

In our everyday life, matter appears smooth and continuous: from the wood used to build your desk to the water you drink. But this is just illusion. The concept that matter is composed of discrete units has been around for millennia. In fact, we know with certainty that our world is composed of microscopic atoms and molecules separated by empty space since the beginning of the twentieth century (Lautrup, 2005). However, certain physical phenomena can be predicted with theories that pay no attention to the molecular structure of materials. Consider for instance the deformation of the horizontal board of a bookshelf under the weight of books. The bending of the shelf can be modelled without considering its molecular composition. The branch of physics in which materials are treated as continuous is known as *continuum mechanics*. Continuum mechanics studies the response of materials to different loading conditions. In this theory, matter is assumed to exist as a continuum, meaning that the matter in the body is continuously distributed and fills the entire region of space it occupies (Lai et al., 1996). This assumption is generally valid if the length scales of interest are large compared with the length scales of discrete molecular structure, but whether the approximation of continuum mechanics is actually justified in a given situation is merely a matter of experimental test.

Modelling anatomical structures requires an understanding of the deformation and stresses caused by the different interactions that occur during medical procedures. A sufficient knowledge of continuum mechanics is therefore essential to follow the rest of this manuscript. Continuum mechanics can be divided into two main parts: general principles common to all media (analysis of deformation, strain and stress concepts) and constitutive equations defining idealised materials. This chapter will not only deal with those two aspects but will also discuss the different aspects of the mechanical behaviour that we find in anatomical structures. Note that this chapter will mostly follow the notation used by Bonnet and Wood (1997) and Reddy (2007). The interested reader may refer to these books for more details.

2.2 Description of motion

Let us consider a body $\boldsymbol{B}$ of known geometry in a three-dimensional Euclidian space $\mathbb{R}^3$. For a given geometry and loading, $\boldsymbol{B}$ will undergo a set of macroscopic changes which is called *deformation*. The region of space occupied by the body at a given time t is termed a *configuration*. A change in the configuration of a continuum body results in a displacement. The displacement of a body has two components: a rigid-body displacement and a deformation. A rigid-body displacement consists of a simultaneous translation and rotation of the body without changing its shape or size. Deformation implies the change in shape and/or size of the body from an initial configuration noted κ_0 to a new configuration κ called the *current* or *deformed configuration*.

Let us now consider a given particle of the body that we call X. What we will call particle in the following is in fact an infinitesimal volume of material. We denote the position it occupies in the initial configuration $\mathbf{X}$ and note its position in the deformed configuration $\mathbf{x}$, both expressed in the chosen frame of reference. The mapping χ defined as the following:

$$\chi : \boldsymbol{B}_{\kappa_0} \to \boldsymbol{B}_{\kappa} \tag{2.1}$$

is called the *deforming mapping* of the body $\boldsymbol{B}$ from κ_0 to κ. When analysing the deformation of a continuous body, it is necessary to describe the evolution of configurations through time. Its mathematical description follows one of the two approaches: the material description or the spatial description. The material description is known as *Lagrangian description* whereas the spatial description is called *Eulerian description*. These two approaches are detailed next.

Olivier Comas - Sat 7$^{\text{th}}$ of August 2010 at 17:15

2.2.1 Lagrangian description

In the Lagrangian description, the position and physical properties of the particles are referred to a reference configuration κ_R, often chosen to be the undeformed configuration κ_0. Thus, the current coordinates ($\mathbf{x} \in \kappa$) are expressed in terms of the reference coordinates ($\mathbf{X} \in \kappa_0$):

$$\mathbf{x} = \chi(\mathbf{X}, t), \quad \chi(\mathbf{X}, 0) = \mathbf{X}, \tag{2.2}$$

and the variation of a typical variable ϕ over the body is described with respect to the coordinates $\mathbf{X}$ and time t:

$$\phi = \phi(\mathbf{X}, t). \tag{2.3}$$

For a fixed value of $\mathbf{X} \in \kappa_0$, $\phi(\mathbf{X}, t)$ gives the value of ϕ at time t associated with the fixed particle X whose position in the reference configuration is $\mathbf{X}$. We note that a change in time t implies that the same particle X has a different value ϕ. Thus, the Lagrangian description focuses its attention on the particles of the continuous body and it is usually used in solid mechanics.

2.2.2 Eulerian description

Rather than following the particles, the Eulerian description observes the changes at fixed locations. In the Eulerian description, the motion is this time referred to the current configuration κ and ϕ is described with respect to the current position ($\mathbf{x} \in \kappa$):

$$\phi = \phi(\mathbf{x}, t), \quad \mathbf{X} = \mathbf{X}(\mathbf{x}, t). \tag{2.4}$$

For a fixed value of $\mathbf{x} \in \kappa$, $\phi(\mathbf{x}, t)$ gives the value of ϕ associated with the particle occupying the position $\mathbf{x} \in \kappa$, which may very well be a different particle for each new time t. Because a change in time t implies that a different value ϕ is observed at the same spatial location $\mathbf{x} \in \kappa$, now probably occupied by a different particle, the Eulerian description is focused on a spatial position. This approach is convenient for the study of fluid flow where the kinematic property of greatest interest is the rate at which change is taking place rather than the shape of the body of fluid at a reference time (Spencer, 1980). Because we are only interested in the study of solid bodies, the Lagrangian description will be used in the rest of the text.

2.2.3 Displacement field

The displacement $\mathbf{u}$ of a particle X is called *displacement vector* and its expression in the Lagrangian description is given by the following:

$$\mathbf{u}(\mathbf{X}, t) = \mathbf{x}(\mathbf{X}, t) - \mathbf{X}. \tag{2.5}$$

A *displacement field* is a vector field of all displacement vectors for all particles in the body, which relates the deformed configuration κ with the undeformed configuration κ_0. Indeed if the displacement field is known, we can construct the current configuration κ from the undeformed configuration κ_0: $\chi(\mathbf{X}, t) = \mathbf{X} + \mathbf{u}(\mathbf{X})$.

2.3 Analysis of deformation

2.3.1 Deformation gradient tensor

The displacement field tells us how a point displaces from the reference to the deformed configuration. However, we would also like to know how a piece of material is stretched and rotated as the body moves from the reference to the deformed configuration. Since the length of a material line $d\mathbf{X}$ can change when going to the deformed configuration as well as its orientation, we can say that $d\mathbf{X}$ deforms into $d\mathbf{x}$. The question then becomes how to relate $d\mathbf{x}$ in the deformed configuration with $d\mathbf{X}$ of the reference configuration.

Consider two particles P_1 and P_2 in a continuous body separated by an infinitesimal distance $d\mathbf{X}$:

$$d\mathbf{X} = \mathbf{X}_{P_2} - \mathbf{X}_{P_1}. \tag{2.6}$$

After deformation, the two particles have deformed to their current positions given by the mapping χ (2.1) as:

$$\mathbf{x}_{P_1} = \chi(\mathbf{X}_{P_1}, t) \quad \text{and} \quad \mathbf{x}_{P_2} = \chi(\mathbf{X}_{P_2}, t). \tag{2.7}$$

Using (2.6) the distance $d\mathbf{x}$ between P_1 and P_2 can then be expressed as:

$$d\mathbf{x} = \mathbf{x}_{P_2} - \mathbf{x}_{P_1} = \chi(\mathbf{X}_{P_1} + d\mathbf{X}, t) - \chi(\mathbf{X}_{P_1}, t). \tag{2.8}$$

The *deformation gradient* $\mathbf{F}$ can be defined as:

$$\mathbf{F} = \frac{\partial \chi}{\partial \mathbf{X}} \tag{2.9}$$

and the vector $d\mathbf{x}$ can then be obtained in terms of $d\mathbf{X}$ as:

$$d\mathbf{x} = \mathbf{F}\, d\mathbf{X}. \tag{2.10}$$

Note that **F** transforms vectors from the reference configuration into vectors in the current configuration and is therefore a second-order tensor.

Knowing that $\chi(\mathbf{X}, t)$ is of course $\mathbf{x}$, the deformation gradient may also be written as:

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \nabla_0 \mathbf{x} = \nabla_0 \mathbf{u} + \mathbf{I}, \tag{2.11}$$

where ∇_0 is the gradient operator with respect to $\mathbf{X}$ and $\mathbf{u}$ the displacement vector. In indicial notation in a Cartesian coordinate system, (2.11) can be explicited as:

$$[F] = \begin{bmatrix} \dfrac{\partial \mathbf{x}_1}{\partial \mathbf{X}_1} & \dfrac{\partial \mathbf{x}_1}{\partial \mathbf{X}_2} & \dfrac{\partial \mathbf{x}_1}{\partial \mathbf{X}_3} \\ \dfrac{\partial \mathbf{x}_2}{\partial \mathbf{X}_1} & \dfrac{\partial \mathbf{x}_2}{\partial \mathbf{X}_2} & \dfrac{\partial \mathbf{x}_2}{\partial \mathbf{X}_3} \\ \dfrac{\partial \mathbf{x}_3}{\partial \mathbf{X}_1} & \dfrac{\partial \mathbf{x}_3}{\partial \mathbf{X}_2} & \dfrac{\partial \mathbf{x}_3}{\partial \mathbf{X}_3} \end{bmatrix}. \tag{2.12}$$

2.3.2 Change of volume

At this point, we have seen how a deformation can affect a vector. We will now look into its effect on volumes. Our motivation comes from the need to write global equilibrium statements that involve integrals over volumes. We can define volume elements in the reference and deformed configurations. Consider three non-coplanar line elements $d\mathbf{X}^{(1)}$, $d\mathbf{X}^{(2)}$ and $d\mathbf{X}^{(3)}$ forming a parallelepiped in the reference configuration TODO: Add figure with all notations . The three vectors after deformation $d\mathbf{x}^{(1)}$, $d\mathbf{x}^{(2)}$ and $d\mathbf{x}^{(3)}$ can be obtained with:

$$d\mathbf{x}^{(i)} = \mathbf{F}\, d\mathbf{X}^{(i)}, \quad i = 1, 2, 3. \tag{2.13}$$

The volume of the parallelepiped that we will note dV can be calculated using the triple product between the three vectors:

$$\begin{aligned} dV &= d\mathbf{X}^{(1)} \cdot d\mathbf{X}^{(2)} \times d\mathbf{X}^{(3)} = (\hat{\mathbf{N}}_1 \cdot \hat{\mathbf{N}}_2 \times \hat{\mathbf{N}}_3)\, dX^{(1)} dX^{(2)} dX^{(3)} \\ &= dX^{(1)} dX^{(2)} dX^{(3)}, \end{aligned} \tag{2.14}$$

where $\hat{\mathbf{N}}_i$ denote the unit vector along $d\mathbf{X}^i$. The corresponding volume in the deformed configuration is given by:

$$\begin{aligned} dv &= d\mathbf{x}^{(1)} \cdot d\mathbf{x}^{(2)} \times d\mathbf{x}^{(3)} \\ &= (\mathbf{F} \cdot \hat{\mathbf{N}}_1) \cdot (\mathbf{F} \cdot \hat{\mathbf{N}}_2) \times (\mathbf{F} \cdot \hat{\mathbf{N}}_3)\, dX^{(1)} dX^{(2)} dX^{(3)} \quad \text{by (2.13)} \\ &= \det \mathbf{F}\, dX^{(1)} dX^{(2)} dX^{(3)}. \end{aligned} \tag{2.15}$$

The determinant of **F** is called the *Jacobian* and it is denoted by $J = \det \mathbf{F}$. And we have:

$$dv = J dV. \tag{2.16}$$

Thus, J has the physical meaning of being the local ratio of current to reference volume of a material volume element.

2.3.3 Change of surface

For similar reasons, let us find the relation between an element of area in the reference configuration $d\mathbf{A}$ which becomes $d\mathbf{a}$ in the deformed configuration. Considering an arbitrary material line $d\mathbf{L}$ in the reference configuration and noting $d\mathbf{l}$ the same line after deformation TODO: Add figure with all notations , the corresponding reference and current volumes are given respectively by:

$$d\mathbf{V} = d\mathbf{L} \cdot d\mathbf{A} \tag{2.17}$$

$$\text{and} \quad d\mathbf{v} = d\mathbf{l} \cdot d\mathbf{a}. \tag{2.18}$$

Using (2.16) which relates the reference and deformed volumes and recalling that $d\mathbf{l} = \mathbf{F}\, d\mathbf{L}$, we have:

$$J\, d\mathbf{L} \cdot d\mathbf{A} = (\mathbf{F}\, d\mathbf{L}) \cdot d\mathbf{a}. \tag{2.19}$$

The fact that this expression is valid for any vector $d\mathbf{L}$ and using the property $\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^T \mathbf{b}$ yields:

$$d\mathbf{a} = J\, \mathbf{F}^{-T}\, d\mathbf{A}. \tag{2.20}$$

2.3.4 Volumetric and isochoric components

We recall that the Jacobian J has the physical meaning of being the local ratio of current to reference volume of a material volume element. Therefore, if $J = 1$ the volume does not change locally during the deformation and the latter is qualified as *isochoric* at this given particle P. If $J = 1$ everywhere in the body, the deformation of the body is isochoric.

When dealing with incompressible and nearly incompressible materials it is necessary to separate the volumetric from the isochoric components of the deformation. Such a separation must ensure that the isochoric component $\hat{\mathbf{F}}$ does not imply any change in volume. This condition can be achieved by choosing $\hat{\mathbf{F}}$ as:

$$\hat{\mathbf{F}} = J^{-1/3}\mathbf{F}. \tag{2.21}$$

Indeed,

$$\begin{aligned} \det \hat{\mathbf{F}} &= \det(J^{-1/3}\mathbf{F}) \\ &= (J^{-1/3})^3 \det \mathbf{F} \\ &= J^{-1} J \\ &= 1. \end{aligned} \tag{2.22}$$

The deformation gradient $\mathbf{F}$ can now be expressed in terms of the volumetric and isochoric component J and $\hat{\mathbf{F}}$ as:

$$\mathbf{F} = J^{1/3}\hat{\mathbf{F}}. \tag{2.23}$$

2.4 Strain measures

The length of a material curve from the reference configuration can change when displaced to a curve in the deformed configuration. If all the curves do not change length, it is said that a rigid body displacement occurred. The concept of strain is used to evaluate how much a given displacement differs locally from a rigid body displacement. Therefore, although we know how to transform vectors from the reference configuration into vectors in the current configuration using the deformation gradient, it is more useful to find a measure of the change in length of $d\mathbf{X}$. Many measures of strains can be defined and the most common ones will now be introduced.

2.4.1 Cauchy-Green deformation tensors

Let us consider two particles P_1 and P_2 in the neighbourhood of each other, separated by $d\mathbf{X}$ in the reference configuration. In the deformed configuration P_1 and P_2 occupy the positions $\tilde{P}_1$ and $\tilde{P}_2$ and they are separated by $d\mathbf{x}$. We are interested in the change of distance between the two points P_1 and P_2 as the body deforms.

The squared distances between P_1 and P_2 and $\tilde{P}_1$ and $\tilde{P}_2$ are respectively given by:

$$\begin{aligned}(dS)^2 &= d\mathbf{X}\cdot d\mathbf{X} &(2.24)\\ (ds)^2 &= d\mathbf{x}\cdot d\mathbf{x} = (\mathbf{F}\,d\mathbf{X})\cdot(\mathbf{F}\,d\mathbf{X})\\ &= (\mathbf{F}\,d\mathbf{X})^T(\mathbf{F}\,d\mathbf{X}) = (d\mathbf{X}^T\,\mathbf{F}^T)(\mathbf{F}\,d\mathbf{X}) = d\mathbf{X}^T(\mathbf{F}^T\mathbf{F}\,d\mathbf{X})\\ &= d\mathbf{X}\cdot(\mathbf{F}^T\mathbf{F}\,d\mathbf{X}). &(2.25)\end{aligned}$$

using the property that the dot product between two vectors a and b can also be expressed as the simple product between the transpose of a and b ($a \cdot b = a^T b$). We define the *right Cauchy-Green deformation tensor* $\mathbf{C}$ as:

$$\mathbf{C} = \mathbf{F}^T\mathbf{F}. \tag{2.26}$$

Thus, the change of distance between the two points P_1 and P_2 after deformation of the continuous body may be written as:

$$(ds)^2 = d\mathbf{X}\cdot(\mathbf{C}\,d\mathbf{X}). \tag{2.27}$$

By definition, $\mathbf{C}$ is a symmetric second-order tensor. The transpose of $\mathbf{C}$ is denoted $\mathbf{B}$ and is called the *left Cauchy-Green deformation tensor*:

$$\mathbf{B} = \mathbf{F}\mathbf{F}^T. \tag{2.28}$$

2.4.2 Green strain tensor

Using (2.24) and (2.27), the change in the squared lengths due to the body deformation between the reference and the current configuration can be expressed as:

$$\begin{aligned}(ds)^2 - (dS)^2 &= d\mathbf{X}\cdot(\mathbf{C}\,d\mathbf{X}) - d\mathbf{X}\cdot d\mathbf{X}\\ &= d\mathbf{X}\cdot(\mathbf{C}-\mathbf{I})\,d\mathbf{X}. &(2.29)\end{aligned}$$

Let us define the *Green-St. Venant strain tensor* $\mathbf{E}$ as:

$$\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I}) \tag{2.30}$$

so we can write:

$$(ds)^2 - (dS)^2 = 2\, d\mathbf{X} \cdot \mathbf{E}\, d\mathbf{X}. \tag{2.31}$$

By definition, the Green strain tensor is a symmetric second-order tensor. Also, the change in squared lengths is zero if and only if $\mathbf{E} = \mathbf{0}$. Using (2.11), the Green strain tensor may be expanded as the following:

$$\begin{aligned}
\mathbf{E} &= \frac{1}{2}(\mathbf{F}^T\mathbf{F} - \mathbf{I}) \\
&= \frac{1}{2}\left[(\nabla_0\mathbf{u} + \mathbf{I})^T(\nabla_0\mathbf{u} + \mathbf{I}) - \mathbf{I}\right] \\
&= \frac{1}{2}\left[(\nabla_0\mathbf{u})^T + \nabla_0\mathbf{u} + (\nabla_0\mathbf{u})^T(\nabla_0\mathbf{u})\right].
\end{aligned} \tag{2.32}$$

The Green strain tensor can be expressed in terms of its components in any coordinate system. In particular, in the Cartesian coordinate system (X_1, X_2, X_3) the components E_{ij} of $\mathbf{E}$ are the following:

$$E_{i,j} = \frac{1}{2}\left(\frac{\partial u_i}{\partial Xj} + \frac{\partial u_j}{\partial Xi} + \frac{\partial u_k}{\partial Xi}\frac{\partial u_k}{\partial Xj}\right), \quad i = 1, 2, 3. \tag{2.33}$$

The components E_{11}, E_{22} and E_{33} are called *normal strains* and it can be shown that they are in fact the ratio of the change in length to the original length along each of the three unit vectors. The components E_{12}, E_{23} and E_{13} are called *shear strains* and they can be interpreted as a measure of the change in angle between line elements that were perpendicular to each other in the undeformed configuration.

2.4.3 Cauchy and Euler tensor

The change in the squared lengths during the body deformation can also be expressed relative to the current length. The length dS can be written in terms of $d\mathbf{x}$ as:

$$(dS)^2 = d\mathbf{X} \cdot d\mathbf{X} = d\mathbf{x} \cdot (\mathbf{F}^{-T}\mathbf{F}^{-1})d\mathbf{x} = d\mathbf{x} \cdot \tilde{\mathbf{B}}d\mathbf{x} \tag{2.34}$$

where $\tilde{\mathbf{B}} = \mathbf{F}^{-T}\mathbf{F}^{-1}$ is called the *Cauchy strain tensor.* $\tilde{\mathbf{B}}$ is in fact the inverse of the left Cauchy-Green tensor $\mathbf{B}$ introduced previously.

In a similar way we defined the Green strain tensor, we can write the change in the squared lengths but relative to the current length:

$$(ds)^2 - (dS)^2 = 2\, d\mathbf{x} \cdot \mathbf{e}\, d\mathbf{x}. \tag{2.35}$$

where $\mathbf{e}$ is called *Almansi-Hamel strain tensor* or simply *Euler strain tensor.*

2.4.4 Principal strains and invariants

The components ε_{ij} of the strain tensor depend on the coordinate system at the point under consideration. However, the strain tensor itself is a physical quantity and as such, it is independent of the coordinate system chosen to represent it. Certain operations on strain tensors give the same result independently of the coordinate system chosen to represent the components of strain. As an example, a vector is a simple tensor of rank one. In three dimensions, it has three components. The value of these components will depend on the coordinate system chosen to represent the vector, but the length of the vector is a physical quantity (a scalar) and is independent of the coordinate system chosen to represent the vector. Similarly, every second rank tensor (such as the strain tensors) has three independent invariant quantities associated with it. One set of such *invariants* are the principal strains of the strain tensor, which are just the eigenvalues of the strain tensor. Because they are independent of any coordinate system, they are very convenient to define strain energy density functions (see section 2.6.5). The most commonly used invariants are:

$$\begin{aligned} I_1 &= tr(\varepsilon) \\ I_2 &= \frac{1}{2}\left\{tr(\varepsilon^2) - [tr(\varepsilon)]^2\right\} \\ I_3 &= det(\varepsilon). \end{aligned} \tag{2.36}$$

It can be shown that in the coordinate system $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$ formed by the three eigenvectors, the expression of the strain tensor is:

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 & 0 & 0 \\ 0 & \varepsilon_2 & 0 \\ 0 & 0 & \varepsilon_3 \end{bmatrix}. \tag{2.37}$$

Since there are no shear strain components in this particular coordinate system, the principal strains represent the maximum and minimum stretches of an elemental volume. Because of the obvious simplicity of the strain tensor's expression, this coordinate system is often used in mechanics to derive and solve equations.

2.4.5 Infinitesimal strain tensor

In some cases, it is possible to simplify the expression of the Green strain tensor defined in (2.32). Indeed, when the displacement gradients are small (that is, $|\nabla\mathbf{u}| \ll 1$) we can neglect the non-linear terms in the definition. In the case of infinitesimal strains, the Green-Lagrange strain tensor and the Eulerian strain tensor are approximately the same and can be approximated by the infinitesimal strain tensor denoted $\boldsymbol{\varepsilon}$ and is given by:

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left[\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right]. \tag{2.38}$$

Its Cartesian components ε_{ij} are the following:

$$\varepsilon_{ij} = \frac{1}{2}(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i}). \tag{2.39}$$

The strain quantities defined in the previous sections are non-linear expressions in terms of the mapping χ and will lead to non-linear governing equations. In solid mechanics, whenever the hypothesis is acceptable, it is common practice to assume that the displacements are small and the infinitesimal strain tensor is then used as a measure of the deformation.

2.5 Stress

Stress is a measure of the intensity of the internal forces acting between particles of a deformable body across imaginary internal surfaces. These internal forces are produced between the particles in the body as a reaction to external forces applied on the body. The *SI* unit for stress is pascal (symbol *Pa*), which is equivalent to one newton (force) per square metre (unit area). As with strain, different measures of stress can be defined. For instance, stress can be measured per unit of deformed area (section 2.5.1) or undeformed area (sections 2.5.2 and 2.5.3). In general, the stress is not uniformly distributed over the cross section of a material body, and consequently the stress at a point on a given area is different from the average stress over this entire area. Therefore, it is necessary to define the stress not over a given area but at a specific point in the body. Stress at a point in a three-dimensional continuum can be measured in terms of nine quantities, three per plane, on three mutually perpendicular planes at the point. These nine quantities may be viewed as the components of a second-order tensor that we call *stress tensor*.

2.5.1 Cauchy stress

The Euler-Cauchy's stress principle states that on any surface (real or imaginary) that divides the body, the action of one part of the body on the other is equivalent to the system of distributed forces and couples on the surface dividing the body. With that consideration in mind, let us consider a plane S that passes through an arbitrary internal point P which has a unit normal vector $\mathbf{n}$. This plane separates the body into two regions and we are interested in the forces applied by one region onto the other. First we will introduce the true stress, defined as the stress in the deformed configuration χ measured per unit area of the deformed configuration. The force $\Delta\mathbf{f}(\mathbf{n})$ acting on a small element of area Δa in a continuous medium depends not only on the size of the area but also depends on the orientation of this area $\mathbf{n}$. The *Cauchy stress vector* at P on S can be defined as:

$$\mathbf{t}(\mathbf{n}) = \lim_{\Delta a \to 0} \frac{\Delta\mathbf{f}(\mathbf{n})}{\Delta a}. \tag{2.40}$$

This equation means that the stress vector depends on its location in the body and the orientation of the plane on which it is acting. In general, the stress vector is not perpendicular to that plane and it can be separated into two components: one component normal to the plane called *normal stress* and one component tangent to the plane named *shear stress*. Since $\mathbf{t}$ depends on $\mathbf{n}$ but is not in the direction of $\mathbf{n}$, it may be interesting to look into the relationship between $\mathbf{t}$ and $\mathbf{n}$.

To establish this relationship, let us consider an infinitesimal tetrahedron in a Cartesian coordinate system. TODO: Add figure of a tetrahedron with all notations . We note $-\mathbf{t}_1$, $-\mathbf{t}_2$, $-\mathbf{t}_3$ and $\mathbf{t}$ the stress vectors in the outwards directions on the faces of the tetrahedron whose areas are respectively Δa_1, Δa_2, Δa_3 and Δa. For the mass m inside the tetrahedron, we have by Newton's second law:

$$\sum \mathbf{F} = \mathbf{t}\Delta a - \mathbf{t}_1\Delta a_1 - \mathbf{t}_2\Delta a_2 - \mathbf{t}_3\Delta a_3 + \Delta v\mathbf{f} = m\mathbf{a}, \tag{2.41}$$

where $\mathbf{f}$ is the force per unit volume acting on the body, Δv the volume of the tetrahedron and $\mathbf{a}$ the acceleration. The areas Δa_1, Δa_2 and Δa_3 being the projections of Δa, they are related to Δa in the Cartesian coordinate system $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ by:

$$\Delta a_i = (\mathbf{n}\cdot\mathbf{e}_i)\Delta a, \quad i = 1, 2, 3. \tag{2.42}$$

Dividing (2.41) by Δa, using (2.42) and noting that $\Delta v/\Delta a \to 0$ when the tetrahedron shrinks to a point yields:

$$\begin{aligned} \mathbf{t} &= (\mathbf{n}\cdot\mathbf{e}_1)\mathbf{t}_1 + (\mathbf{n}\cdot\mathbf{e}_2)\mathbf{t}_2 + (\mathbf{n}\cdot\mathbf{e}_3)\mathbf{t}_3 \\ &= \mathbf{n}\cdot(\mathbf{e}_1\mathbf{t}_1 + \mathbf{e}_2\mathbf{t}_2 + \mathbf{e}_3\mathbf{t}_3). \end{aligned} \tag{2.43}$$

We define the *Cauchy stress tensor* $\sigma = \mathbf{e}_1\mathbf{t}_1 + \mathbf{e}_2\mathbf{t}_2 + \mathbf{e}_3\mathbf{t}_3$ and we have:

$$\mathbf{t}(\mathbf{n}) = \mathbf{n}\cdot\sigma. \tag{2.44}$$

The stress vector $\mathbf{t}$ represents the vectorial stress on a plane whose normal is $\mathbf{n}$. As we demonstrated, the stress tensor is a property of the medium that is independent of $\mathbf{n}$. It represents the current force per unit of deformed area: $d\mathbf{f} = \mathbf{t}da = \sigma\cdot d\mathbf{a}$.

The matrix form of (2.44) in a Cartesian coordinate system is given by:

$$\begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} = \begin{bmatrix} \sigma_{11} & \sigma_{21} & \sigma_{31} \\ \sigma_{12} & \sigma_{22} & \sigma_{32} \\ \sigma_{13} & \sigma_{23} & \sigma_{33} \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix}. \tag{2.45}$$

The Cauchy stress tensor is used for stress analysis of material bodies experiencing small deformations. This stress is basically defined as force/(unit deformed area). The strain measure that is appropriate to use with the Cauchy stress tensor is the infinitesimal strain tensor (see section 2.4.5). The problem with using the Cauchy stress tensor for analysing materials undergoing large deformation is that we generally do not know the area in the deformed configuration. Therefore, we need to develop alternative stress tensors that we can use in the reference configuration. We will introduce two new stress measures: the first Piola-Kirchhoff stress tensor and the second Piola-Kirchhoff stress tensor.

2.5.2 First Piola-Kirchhoff stress tensor

We know that the Cauchy stress tensor represents the current force per unit of deformed area: $d\mathbf{f} = \mathbf{t}(\mathbf{n})da = \sigma \cdot d\mathbf{a}$. We define a stress vector $\mathbf{T}$ over the area element dA with normal $\mathbf{N}$ in the underformed configuration such that it results in the same total force

$$d\mathbf{f} = \mathbf{t}(\mathbf{n})da = \mathbf{T}(\mathbf{N})dA. \tag{2.46}$$

The two stress vectors have the obviously the same direction but different magnitudes since they are not applied to the same area.

In a similar way we defined the Cauchy stress σ with $\mathbf{t}(\mathbf{n}) = \sigma \cdot \mathbf{n}$, we introduce a stress tensor $\mathbf{P}$ called the *first Piola-Kirchhoff stress tensor* such that $\mathbf{T}(\mathbf{n}) = \mathbf{P} \cdot \mathbf{N}$. The first Piola-Kirchhoff stress tensor gives the current force per unit undeformed area.

The relation between the first Piola-Kirchhoff stress tensor and the Cauchy stress tensor can be obtained as follows. From (2.46) we have:

$$\mathbf{T} = \frac{da}{dA}\mathbf{t}. \tag{2.47}$$

Using $d\mathbf{f} = \mathbf{t}da$ and the relation (2.20) between $d\mathbf{A}$ and $d\mathbf{a}$, we have:

$$\mathbf{T} = J\,\mathbf{t}\,\mathbf{F}^{-T}. \tag{2.48}$$

Finally we can express the first Piola-Kirchhoff stress tensor $\mathbf{P}$ as

$$\mathbf{P} = J\,\sigma\,\mathbf{F}^{-T}. \tag{2.49}$$

2.5.3 Second Piola-Kirchhoff stress tensor

The *second Piola-Kirchhoff stress tensor* $\mathbf{S}$ is introduced as the stress tensor associated with the force $d\mathbf{F}$ in the undeformed area $d\mathbf{A}$ that corresponds to the force $d\mathbf{f}$ in the deformed area $d\mathbf{a}$:

$$d\mathbf{F} = \mathbf{S}\,d\mathbf{A}. \tag{2.50}$$

Thus, the second Piola-Kirchhoff stress tensor gives the transformed current force per unit undeformed area. Similar to the relationship (2.13) between $d\mathbf{x}$ and $d\mathbf{X}$, the force $d\mathbf{f}$ is related to the force $d\mathbf{F}$ by:

$$\begin{aligned} d\mathbf{F} &= \mathbf{F}^{-1}\,d\mathbf{f} \\ &= \mathbf{F}^{-1}\,(\mathbf{P}\,d\mathbf{A}) \\ &= \mathbf{S}\,d\mathbf{A}. \end{aligned} \tag{2.51}$$

Therefore, the second Piola-Kirchhoff stress tensor is related to the first Piola-Kirchhoff stress tensor as follows:

$$\mathbf{S} = \mathbf{F}^{-1}\mathbf{P} = J\,\mathbf{F}^{-1}\,\sigma\,\mathbf{F}^{-T}. \tag{2.52}$$

2.5.4 Principal stresses and invariants

In the same way that we defined principal strains in section 2.4.4 and like any second-rank tensor, the stress tensor has at least three invariants and we can find a coordinate system $(\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3)$ in which the expression of the stress tensor is:

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}. \tag{2.53}$$

This result means that at every point in a stressed body there are at least three planes called *principal planes*, with normal vectors $\mathbf{n}_1$, $\mathbf{n}_2$ and $\mathbf{n}_3$ called *principal directions* (in fact, the eigenvectors), where the corresponding stress vector is perpendicular to the plane and where there are no normal shear stresses. The three stresses normal σ_1, σ_2 and σ_3 to these principal planes are called *principal stresses.*

2.6 Constitutive equations

So far, all the relations that we have established are valid for every continuum. Indeed, no mention was made of any material in the derivations. Therefore, these relations are not sufficient to describe the response of a material to a given loading. We know that the stresses result from the deformation of the material and it is now necessary to express them in terms of some measure of this deformation such as, for instance, the strain. These relationships, known as *constitutive equations*, obviously depend on the type of material under consideration and may be dependent on time or not. Constitutive equations are not derived from any physical principles, although they are subject to obeying certain rules and the entropy inequality. In essence, constitutive equations are mathematical models of the behaviour of materials that are validated against experimental results.

The relationship between stress and strain depends on the nature, or constitution of the particular material that we are studying. Of course, any material mechanics, and especially tissue mechanics, can be quite complex, and many assumptions must be made when deriving a constitutive equation. It is basically impossible to derive a particular constitutive equation that would accurately model all aspects of tissue behaviour under any type of loading. This is because tissues may behave very differently under small loads as compared to large loads. For instance, we may expect damage under large loads. Also, under different loading rates different tissues may behave differently. Therefore, whenever we develop a constitutive equation to model a tissue, we need to balance the need to accurately model tissue behaviour under the range of loadings with the need to have a constitutive equation that is simple enough to:

1. be computed in a numerical model
2. experimentally measure all the material constants used in the constitutive equation

Olivier Comas - Sat 7$^{\text{th}}$ of August 2010 at 17:15

This section will now introduce the most common constitutive models relevant to soft-tissue modelling.

2.6.1 Elasticity

If under applied loads a material stores but does not dissipate energy, and it returns to its orginal shape when the loads are removed, we call that material *elastic*. The work done by the stress is generally dependent on the deformation path. However, for elastic materials the stress does not depend on the path of deformation and therefore the state of stress in the deformed configuration is determined only by the state of deformation. It means that the stress does not depend on the time taken to achieve the deformation or the rate at which the state of deformation is reached.

One special case of elastic materials are *hyperelastic* materials. A material is said to be hyperelastic if there exists a *strain energy density* function $U_0(\varepsilon)$ such that

$$\boldsymbol{\sigma} = \frac{\partial U_0}{\partial \varepsilon}. \tag{2.54}$$

However, for *imcompressible* materials the stress tensor is not completely determined by the deformation. The hydrostatic pressure[1] affects the stress and the above equation is then written as:

$$\boldsymbol{\sigma} = -p\mathbf{I} + \frac{\partial U_0}{\partial \varepsilon}, \tag{2.55}$$

where p is the hydrostatic pressure. In order to develop a mathematical model of an hyperelastic material, U_0 is expanded in Taylor's series about $\varepsilon = 0$:

$$U_0 = C_0 + C_{ij}\varepsilon_{ij} + \frac{1}{2!}\hat{C}_{ijkl}\varepsilon ij\varepsilon_{kl} + \frac{1}{3!}\hat{C}_{ijklmn}\varepsilon_{ij}\varepsilon_{kl}\varepsilon_{mn} + \ldots, \tag{2.56}$$

where C_0, C_{ij} and $\hat{C}$ are material stiffnesses. The form of U_0 may varied for different materials. A material is said to be *linear* if the relationship between strain and stress is also linear. Based on this definition, U_0 is a cubic or higher-order function of the strains for non-linear materials whereas U_0 is a quadratic function of strain for linear materials. We will start by discussing linear materials for the case of infinitesimal deformations (sections 2.6.2 to 2.6.4). Consequently, we will not make any distinction between the various measures of stress and strain and use $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$ for stress and strain tensors respectively.

2.6.2 Linear materials: generalised Hooke's law

The linear constitutive model for infinitesimal deformations is called the generalised Hooke's law. To derive the relation between stress and strain for a linear material, we assume the quadratic form of U_0:

$$U_0 = C_0 + C_{ij}\varepsilon_{ij} + \frac{1}{2!}\hat{C}_{ijkl}\varepsilon ij\varepsilon_{kl}. \tag{2.57}$$

---

[1] pressure exerted by a fluid at equilibrium due to the force of gravity

From there and using (2.54) it can be shown that there exists some coefficients C_{ijkl} such as:

$$\sigma_{ij} = C_{ijkl}\varepsilon_{kl}. \tag{2.58}$$

The C_{ijkl} are the coefficients of the fourth-order tensor $\mathbf{C}$ that relates the second-order tensors of strain and stress. The 81 scalar components of this fourth-order tensor ($= 3^4$) can be reduced to 21 independent coefficients using existing symmetries. Recalling that both strain and stress tensors are symmetric yields:

$$\begin{bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \sigma_{23}\\ \sigma_{13}\\ \sigma_{12}\end{bmatrix} = \begin{bmatrix} C_{1111} & C_{1122} & C_{1133} & C_{1123} & C_{1113} & C_{1112}\\ C_{1122} & C_{2222} & C_{2233} & C_{2223} & C_{2213} & C_{2212}\\ C_{1133} & C_{2233} & C_{3333} & C_{3323} & C_{3313} & C_{3312}\\ C_{1123} & C_{2223} & C_{3323} & C_{2323} & C_{2313} & C_{2312}\\ C_{1113} & C_{2213} & C_{3313} & C_{2313} & C_{1313} & C_{1312}\\ C_{1112} & C_{2212} & C_{3312} & C_{2312} & C_{1312} & C_{1212}\end{bmatrix} \begin{bmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{33}\\ \varepsilon_{23}\\ \varepsilon_{13}\\ \varepsilon_{12}\end{bmatrix}. \tag{2.59}$$

2.6.3 Orthotropic materials

Further reduction in the number of independent stiffness parameters can be achieved using the so-called material symmetry. This symmetry is due to the internal structure of the material, that is the arrangement of molecules (crystallographic form). We note that the symmetry under discussion is a directional property and not a positional property. Thus, a material may have certain elastic symmetry at every point of a material body and the properties may vary from point to point. Positional dependence of material properties is what we called the inhomogeneity of the material.

Orthotropic materials are characterised by three mutually orthogonal planes of material symmetry, that is, three mutually orthogonal preferred directions. In this case, the number of coefficients is reduced to 9 and the stress-strain relation for an orthotropic material is the following:

$$\begin{bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \sigma_{23}\\ \sigma_{13}\\ \sigma_{12}\end{bmatrix} = \begin{bmatrix} C_{1111} & C_{1122} & C_{1133} & 0 & 0 & 0\\ C_{1122} & C_{2222} & C_{2233} & 0 & 0 & 0\\ C_{1133} & C_{2233} & C_{3333} & 0 & 0 & 0\\ 0 & 0 & 0 & C_{2323} & 0 & 0\\ 0 & 0 & 0 & 0 & C_{1313} & 0\\ 0 & 0 & 0 & 0 & 0 & C_{1212}\end{bmatrix} \begin{bmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{33}\\ \varepsilon_{23}\\ \varepsilon_{13}\\ \varepsilon_{12}\end{bmatrix}. \tag{2.60}$$

Most often, the material properties are determined in a laboratory in terms of the engineering constants such as Young's modulus, shear modulus, and so on. These constants are measured using simple tests like uniaxial tension test or pure shear test. Because of their direct and obvious physical meaning, engineering constants are used in place of the more abstract stiffness coefficients C_{ijkl}. Therefore, we will now reformulate (2.60) with respect to those engineering constants. Noting E_i the

Young's modulus in the direction x_i and G_{ij} the shear moduli in the $x_i - x_j$ plane, it can be shown that:

$$\begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \varepsilon_{23} \\ \varepsilon_{13} \\ \varepsilon_{12} \end{bmatrix} = \begin{bmatrix} \frac{1}{E_1} & -\frac{\nu_{21}}{E_2} & -\frac{\nu_{31}}{E_3} & 0 & 0 & 0 \\ -\frac{\nu_{12}}{E_1} & \frac{1}{E_2} & -\frac{\nu_{32}}{E_3} & 0 & 0 & 0 \\ -\frac{\nu_{13}}{E_1} & -\frac{\nu_{23}}{E_2} & \frac{1}{E_3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{23}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{13}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{12}} \end{bmatrix} \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix}. \qquad (2.61)$$

2.6.4 Isotropic materials

A material is said *isotropic* if its properties are independent of directions. Therefore, we have the following relations:

$$E_1 = E_2 = E_3 = E, \quad G_{12} = G_{13} = G_{23} = G, \quad \nu_{12} = \nu_{23} = \nu_{13} = \nu. \qquad (2.62)$$

The stress-strain relations may be written:

$$\sigma_{ij} = \frac{E}{1+\nu}\varepsilon_{ij} + \frac{\nu E}{(1+\nu)(1-2\nu)}\varepsilon_{kk}\delta_{ij}. \qquad (2.63)$$

where the summation of repeated indices is implied (Einstein's notation) and δ_{ij} is the Kronecker delta[2]. The inverse relations are:

$$\varepsilon_{ij} = \frac{1+\nu}{E}\sigma_{ij} - \frac{\nu}{E}\sigma_{kk}\delta_{ij}, \qquad (2.64)$$

recalling that $G = E/[2(1+\nu)]$.

2.6.5 Non-linear materials

For most materials, the linear relation between stress and strain is only valid for small deformation. Beyond a certain threshold, Hooke's law is no longer valid. Note that it does mean that the material is not elastic, it may well recover all its deformation after removal of the loads. However, materials often exibit an elastic threshold in the non-linear range after which permanent deformation occurs. This range of deformation is qualified of *plastic*.

The strain energy density function takes different forms for different materials. It is often expressed as a linar combination of unknown coefficients (determined experimentally) and principal invariants of Green strain tensor **E**, left Cauchy-Green strain tensor **B** and deformation gradient tensor **F** (see section 2.4.4 for more information on invariants). The constitutive equations of two well-known non-linear elastic materials (Mooney-Rivlin and neo-Hookean) will now be introduced.

---

[2] δ_{ij} is equal to 1 if $i = j$ and to 0 if $i \neq j$

Mooney-Rivlin

The Mooney-Rivlin model was proposed by Melvin Mooney and Ronald Rivlin in two independent papers in 1952. For an incompressible Mooney-Rivlin material, the strain energy density function U_0 is taken as a linear function of the principal invariants of the left Cauchy-Green strain tensor $\mathbf{B}$ and is written as:

$$U_0 = C_1(I_B - 3) + C_2(II_B - 3), \tag{2.65}$$

where C_1 and C_2 are two constants and I_B and II_B are the two principal invariants of $\mathbf{B}$. For those materials, it can be shown that the stress tensor has the form

$$\sigma = -p\mathbf{I} + \alpha\mathbf{B} + \beta\mathbf{B}^{-1}, \tag{2.66}$$

where ρ is the material density and α and β are given by

$$\alpha = 2\rho\frac{\partial U_0}{\partial I_B} = 2\rho C_1, \quad \beta = -2\rho\frac{\partial U_0}{\partial II_B} = -2\rho C_2. \tag{2.67}$$

The Mooney-Rivlin incompressible model is often used to represent the behaviour of rubber-like materials.

Neo-Hookean

The neo-Hookean model was proposed by Ronald Rivlin in 1948. It can be derived from the Mooney-Rivlin model in the special case where $C_2 = 0$ in the strain energy density function. Therefore, we have:

$$U_0 = C_1(I_B - 3). \tag{2.68}$$

and the constitutive equation for the stress may be written as:

$$\sigma = -p\mathbf{I} + 2\rho C_1\mathbf{B}. \tag{2.69}$$

The neo-Hookean model provides a reasonable approximation of rubber-like material for moderate strains. It is typically accurate for strains less than 20 % (Gent, 2001).

2.6.6 Viscoelastic materials

Viscoelasticity is the property of materials that exhibit both viscous and elastic characteristics when undergoing deformation. Viscous materials, like honey, resist shear flow and strain linearly with time when a stress is applied. Elastic materials strain instantaneously when stretched and just as quickly return to their original state once the stress is removed. Viscoelastic materials have elements of both of these properties and, as such, exhibit time dependent strain. Depending on the change of strain rate versus stress inside a material the viscosity can be categorised as having a linear or non-linear response. When a material exhibits a linear response it is categorised as a Newtonian material. In this case the stress is linearly proportional to the strain rate. If the material exhibits a non-linear response to the strain rate, it is categorised as a Non-Newtonian material.

Viscoelastic materials present a few characteristics:

1. if the stress is held constant, the strain increases with time (creep)
2. if the strain is held constant, the stress decreases with time (relaxation)
3. the effective stiffness depends on the rate of application of the load
4. if cyclic loading is applied, hysteresis occurs, leading to a dissipation of mechanical energy

TODO: Add details on constitutive equations? Level of details to be determined later (depending on TLED needs)

2.7 Tissue characterisation

One of the main obstacles in developing realistic organ models is the lack of data on the material properties of live organ tissues. Measuring and characterising in vivo organ properties is a highly challenging task, but is a requirement for realistic organ modelling. Indeed, as we have seen in sections 2.60 and 2.6.5, constitutive equations require the experimental determination of material constants. Organ models with incorrect material properties will affect training in surgical simulator systems (Sedef et al., 2006).

The constitutive equations are used to define idealised materials, which represent different aspects of the mechanical behaviour of natural materials. However, no real material is known to behave exactly as one of these mathematical models. In particular, few biological tissues obey Hooke's law. If because of their high stiffness (and therefore small deformations) cortical bones are accurately described by a linear model, Hooke's law becomes insufficient to describe the mechanical behaviour of the liver with precision for instance. In fact, in vivo experiments conducted by Melvin et al. (1973) suggest that mechanical behaviour of liver is non-linear and viscoelastic. The brain also appears to be non-linear and viscoelastic (Miller and Chinzei, 1997). The liver has the additional property of anisotropy, more precisely, it is transversely isotropic according to Chui et al. (2007).

TODO: Transition with next chapter (complexity of models => FEM to solve them)

CHAPTER 3

PRACTICAL APPROACH OF THE FINITE ELEMENT METHOD

Blabla

3.1 Introduction

3.1.1 A numerical method

One of the most important things engineers and scientists do is to model physical phenomena. Using assumptions concerning how the phenomena works and using the appropriate laws of physics governing the process, they can derive a mathematical model, often characterised by complex differential or integral equations relating various quantities of interest. However, because of the complexities in the geometry and complex boundary conditions found in real life problems, we cannot analytically solve these equations. A few decades ago, the only possible approach was to drastically simplify them, which was not always sufficient to find an approximate solution. Nowadays, in practice, most of the problems are solved using numerical methods. Indeed, with suitable mathematical models and numerical methods, computers can help solving many practical problems of engineering. Numerical methods typically transform differential equations governing a continuum to a set of algebraic equations of a discrete model of the continuum that are to be solved using computers (Reddy, 1993). The *Finite Element Method* (FEM) is the most popular numerical procedure that is used to approximately solve differential equations, especially in continuum mechanics.

3.1.2 The basic ideas of FEM

The finite element method begins by dividing the structure into small pieces, manageable regions, called *elements*. The collection of all these elements makes up a *mesh* which approximates the problem geometry. Why is this a good idea? If we can expect the solution for an engineering problem to be very complex, it is mostly due to the complex geometry, on which a global solution is difficult to figure out. If the problem domain can be divided (*meshed*) into small elements, the solution within an element is easier to approximate (MacDonald, 2007). Over each finite element, the unknown variables are approximated using known functions. These functions can be linear or higher-order polynomial expressions that depend on the geometrical locations (*nodes*) used to define the finite element shape. The governing equations are integrated over each finite element using linear algebra techniques and the solution over the entire domain problem is obtained by summing (*assembling*) the solution of each element. Thus, the finite element method transforms an infinite number of differential equations (one can be defined at any point of the continuum) into a finite number of algebraic equations (depending on the chosen number of elements).

This approach may be compared to trying to find the area under a curve. We know that we can find the exact solution for the area under the curve by integration. However, sometimes the function describing the curve is not known, or is difficult to integrate. One method to obtain an approximate solution is to break up the area into a series of rectangles and add the areas of all rectangles TODO: Add figure with curve and rectangles . It is worth nothing that the solution accuracy can be increased by reducing the width of the rectangles to better follow the curve.

It is crucial to keep in mind that approximations occur at different stages during finite element analysis. The division of the whole domain into finite elements may not be exact TODO: add figure from a domain divided into elements , introducing error in the domain being modelled. The second stage is when element equations are derived. As mentionned earlier, the unknowns of the problem are approximated using the idea that any continuous function can be represented by a linear combination of known functions and undetermined coefficients. Algebraic relations between the undetermined coefficients are then obtained by satisfying the governing equations over each element. There are a few types of approach for establishing these equations but, without going into details, the mathematical foundations of all these approaches are energy principles or the *weighted residual methods* (see Appendix B), which both lead to integrals during the process. Therefore, the second stage creates two sources of error: the representation of the solution by a linear combination of functions and the evaluation of the integrals. Finally, errors are introduced when solving the assembled system of algebraic equations.

3.2 Discretisation

3.2.1 Meshing process

The first step in the finite element method is to create a mesh of the domain to study. Mesh generation is a very important task and can be very time consuming. The domain has to be meshed properly into elements of specific shapes. All the elements together form the entire domain of the problem without any gap or overlapping. For example, triangles or quadrilaterals can be used in two dimensions, and tetrahedra and hexahedra in three dimensions. Information, such as *element connectivity*, must be created during the meshing process for later use in the formulation of the FEM equations. The number of elements into which the domain is divided in a problem depends mainly on the geometry of the domain and on the desired accuracy of the solution. Usually, the number of elements increases with the complexity of the geometry. For instance, if one part of the domain is thiner TODO: add figure to illustrate this point , the size of the elements must be reduced in order to tile this particular part, which increases the total number of elements, and hence the number of algebraic equations to solve. However, adding elements is sometimes desirable. Indeed, increasing the number of elements tends to get an approximate solution closer to the exact analytical solution (as reducing the width of the rectangles allows for a better approximation of the area under the curve). Elements are classically added in the particular regions of interest for a given problem and where substantial deformation may occur if this information is known a priori. A trade-off between accuracy and computational time must be found. This is why the meshing process on the problem domain must be carried out carefully. One needs to create a mesh which gives an accurate enough solution for the desired application while restraining the computational time according to the problem time constraints. As an example, finding the maximum load that can sustain a bridge in structural mechanics demands a high degree of accuracy, no matter how much time it is needed to compute the solution. Conversely, organ deformation in medical training simulators must be computed at an interactive rate so that no apparent delay can be observed between the manipulation of a given organ and its deformation. It does not mean that precision is not required, simply that the time constraints will necessarily limit the accuracy of the solution.

3.2.2 Solution interpolation

As we have seen, the finite element method is based on finding an approximate solution over each simple element rather than the whole domain. Any continuous function f may be approximated by a linear combination of known functions ϕ_i and undetermined coefficients c_i:

$$f \approx \tilde{f} = \sum_{i=1}^{n} c_i \phi_i. \tag{3.1}$$

Moreover, the approximation solution u^ewithin the element Ω_e must fullfill certain conditions of continuity in order to be convergent to the actual solution u as the number of elements is increased. The finite element method approximates the solution by the following polynomial expression:

$$u^e(x) = \sum_{i=1}^{n} u_i^e \psi_i^e(x) \tag{3.2}$$

where Ω_e is a one-dimensional element[1], x a position within this element, u_i^e are the values of the solution u at the nodes and $\psi_i^e(x)$ the approximation functions over the element. This particular form will be assumed for brevity but the interested reader may refer to Reddy (1993) for demonstration. Note that u_i^e plays the role of the undetermined coefficients and ψ_i^e the role of approximation functions. Writing the approximation in terms of the nodal values of the solution is necessitated by the fact that we require the solution to be the same at points common to the elements in order to connect the approximate solution from each element and form a continuous solution over the whole domain. The approximation ψ_i^e can be linear or higher-order polynomial expressions and are called interpolation functions. Depending on the degree of polynomial approximation used to represent the solution, additional nodes may be identified inside the element. It is worth noting that the type of interpolation directly affects the accuracy of the solution. In other words, finding the solution to the problem consists of merely figuring out the values of the sought variable at every node of the mesh. Its value at any other point of the domain may then be deducted by interpolation within elements.

---

[1] For the sake of simplicity, all derivations are carried out in 1D but remain valid for each component in higher dimensions.

The next logical question is how to derive the interpolation functions ψ_i^e for a given element. Their derivation depends only on the geometry of the element and the number and location of the nodes. The number of nodes must be equal to the number of terms in the polynomial expansion. Therefore each element contains a single interpolation function for each of its nodes. As stated by interpolation theory, each interpolation function is required to be equal to 1 at its corresponding node, and 0 at all other nodes:

$$\psi_i^e(x_j) = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{3.3}$$

where x_j is the position of node j. From there, we can state that the interpolation function at node i may be written as:

$$\psi_i^e(x) = C_i \prod_{i \neq j}^{n-1} (x - x_j) \tag{3.4}$$

where j are the indices of the other nodes and C_i is a constant to be determined such as:

$$\psi_i^e(x_i) = 1. \tag{3.5}$$

This function is zero at all nodes except the i^{th} node. It is worth noting that such a definition for the interpolation functions ψ_i^e used in (3.2) yields:

$$u^e(x_j) = \sum_{i=1}^{n} u_i^e \psi_i^e(x_j) = u_j^e \tag{3.6}$$

as expected.

Since elements over the whole domain are generally not identical (different shapes), this process of determining all interpolation functions can become fastidious. We will now introduce a general method for defining interpolation functions which allows for arbitrary element shapes (subject to a given topology). To this end, we discuss the concept of element *natural coordinates*.

3.2.3 Natural coordinates

The natural coordinate system allows us to map every element into a typical and simpler element. As an example, an 8-node hexahedron element is shown in Fig. TODO: add figure of hexahedron in global and natural coordinates as it appears in the global (x_1, x_2, x_3) and natural (ξ_1, ξ_2, ξ_3) coordinate systems. While in its natural coordinate system the element is a regular aligned cube, in the global coordinate system the element may assume any admissible arbitrary form. Essentially, admissible means that the element must not be too distorted, and must certainly not fold back on itself. Another example is given Fig. TODO: add figure of tetrahedron in global and natural coordinates with a tetrahedral element. The components of each node has a simple expression in natural coordinates, usually $0, 1$ or -1 and the centre of the coordinate system is taken at one of the nodes or at the centre of the element.

The use of the natural coordinate system has several advantages (Biswas et al., 1976). Not only the interpolation functions can be derivated only once per type of element in the mesh, but their expression is also much simpler, regardless of the actual element shape. Consequently, the element equations and the derived element matrices get to be simplified too. However, we now need to switch between the natural and global coordinate systems before solving the equations on the whole domain. Indeed, it is necessary to account for the diversity of element shapes within the mesh. Solving the element equations expressed into the natural coordinate system would not yield the expected result for the whole domain.

3.2.4 Geometry interpolation

Let us assume a relation between the global coordinate x and the natural coordinate ξ in the following form:

$$x = f(\xi) \tag{3.7}$$

where f is assumed to be a one-to-one correspondence (that is, a bijective function). This function may be seen as a transformation between the natural shape of the element and its arbitrary shape within the actual mesh, it describes a change in geometry. It is natural to think of approximating the geometry in a similar way that we approximated the solution in section 3.2.2. Hence, the transformation defined by (3.7) may also be written as

$$x = \sum_{i=1}^{m} x_i^e \hat{\psi}_i^e(\xi) \tag{3.8}$$

where x_i^e is the global coordinate of the i^{th} node of the element Ω_e, m the number of nodes for the element and $\hat{\psi}_i^e(\xi)$ are the interpolation functions of degree $m-1$. Thus, we have a linear transformation when $m = 2$ and the relation betweeen x and ξ is quadratic when $m = 3$. The interpolation functions $\hat{\psi}_i^e(\xi)$ are called *shape functions* because they are used to express the geometry or shape of the element.

Remark. It is worth noting that (3.8) can be easily extented to three-dimensional problems. Let us consider a Cartesian coordinate system and we denote the position of node i $\mathbf{x}_i = [x_{i1}, x_{i2}, x_{i3}]^T$. The position of a point $\mathbf{x} = [x_1, x_2, x_3]^T$ within the element may then be interpolated by applying (3.8) to each component of the position:

$$\begin{aligned} x_1 &= \sum_{i=1}^{m} x_{i1}\hat{\psi}_i^e(\xi) \\ x_2 &= \sum_{i=1}^{m} x_{i2}\hat{\psi}_i^e(\xi) \\ x_3 &= \sum_{i=1}^{m} x_{i3}\hat{\psi}_i^e(\xi) \end{aligned} \tag{3.9}$$

If we construct the vector $\hat{\mathbf{x}}^e$ of all nodal positions $\mathbf{x}_i$ of the element, we can define a shape function matrix $\mathbf{H}$ such as:

$$\mathbf{x} = \mathbf{H}\hat{\mathbf{x}}^e \tag{3.10}$$

where $\mathbf{H}$ is a $3 \times m$ matrix built from submatrices $\mathbf{H}_i$

$$\mathbf{H}_i = \begin{bmatrix} \hat{\psi}_i^e & 0 & 0 \\ 0 & \hat{\psi}_i^e & 0 \\ 0 & 0 & \hat{\psi}_i^e \end{bmatrix} \quad i = 1, 2, \ldots, m. \tag{3.11}$$

3.2.5 A particular case: isoparametric elements

Most of the time, we choose to interpolate the solution and the element geometry with the same interpolation functions. In this case, the elements are said to be *isoparametric*. Under this configuration, natural coordinates appear as parameters that define the shape functions. Note that the shape functions describe the geometry and the solution with the same degree of interpolation. Because isoparametric elements are very common in practice, the use of the phrase *shape functions* is often extented to denote the interpolation functions employed to approximate the solution.

3.3 Derivation of element equations

Let us sum up where we are into the finite element analysis process so far. The whole domain has been divided into sub-domains, smaller, that we call elements, for which the solution to be sought will be simpler to find. We know how to approximate the solution within each of these elements using a linear combination of shape functions. And we even have a method to simplify the expression of these shape functions using natural coordinates.

3.3.1 Strong and weak forms

The next step is therefore to derive the element equations themselves. Obviously these equations depend on the type of problem that we seek to solve. From now on, we will only consider the context of continuum mechanics since our overall goal is the modelling of organ deformation. However, a similar approach can be used in the other fields of physics. In continuum mechanics, applying Newton's second law yields a partial differential system of equations. Such equations are called *strong forms*. The strong form, in contrast to a *weak form*, requires strong continuity on the dependent field variables, such as displacements (Liu and Quek, 2003). The functions defining these field variables have to be differentiable up to the order of the partial differential equations that exist in the strong from. Obtaining the exact solution for a strong form is usually very difficult for practical engineering problems. We know that the finite element method can be used to find an approximated solution but the method usually works well only with regular geometry and boundary conditions.

A weak form is often created using energy principles or weighted residual methods. The weak form is often an integral form and requires a weaker continuity on the field variables. Due to the weaker requirement on the field variables, and the integral operation, a formulation based on a weak form usually produces a set of discretised system equations that give much more accurate results, especially for problems of complex geometry. Hence, the weak form is preferred by many for obtaining an approximate solution. Using the weak form usually leads to a set of well-behaved algebraic system equations, if the problem domain is discretised properly into elements.

3.3.2 Time dependence

In all the equations of the finite element method that we derived so far, we have not taken time into account. For some problem, this is not an issue, the motion is sufficiently slow to assume that the dynamic effects (such as damping effect) are neglectable. However, this assumption is not always valid, particulary in our case where we want to model organ deformation. The deformation obviously depends on time. Finite element models of time-dependent problems can be developed in two alternatives ways (Reddy, 1993):

(a) coupled formulation in which the time t is treated as an additional coordinate along with the spatial coordinate x

(b) decoupled formulation where time and spatial variations are assumede to be separable.

Thus, the approximation of the solution u takes one of these two forms:

$$u(x,t) \approx u^e(x,t) = \sum_{i=1}^{n} \hat{u}_i^e \hat{\psi}_i^e(x,t) \quad \text{(coupled formulation)} \tag{3.12}$$

$$u(x,t) \approx u^e(x,t) = \sum_{i=1}^{n} u_i^e(t) \psi_i^e(x) \quad \text{(decoupled formulation)} \tag{3.13}$$

where $\hat{\psi}_i^e(x,t)$ are time-space interpolation functions, $\hat{u}_i^e$ are the nodal values that are independent of x and t, $\psi_i^e(x)$ are the usual interpolation functions in spatial coordinate x only and the nodal values $u_i^e(t)$ are functions of time t only. Space-time coupled finite element formulations are not common and they have not been adequately studied. Hence, we consider the decoupled formulation only. Of course, the assumption that the time and spatial variations are separable is not valid in general. However, with sufficiently small time steps, it is possible to obtain accurate solutions nevertheless.

3.3.3 Dynamic system of equations

The space-time decoupled finite element formulation of time-dependent problems involves two steps:

1. **Spatial approximation.** The solution u is first approximated by expressions of the form (3.13) while keeping the time-dependent term during the finite element model derivation. This step leads to a set of ordinary differential equations in time.

2. **Temporal approximation.** The system of ordinary differential equations is then approximated in time where different schemes may be used to assess the time derivatives. This step yields a set of algebraic equations for u_i^e at discrete time $t_{k+1} = (k+1)\Delta t$ where Δt is the time increment and $k > 0$ an integer.

At the end of this two-step approximation, we have a continuous spatial solution at discrete intervals in time:

$$u(x, t_k) \approx u^e(x, t_k) = \sum_{i=1}^{n} u_i^e(t_k)\psi_i^e(x) \quad \text{with } k = 0, 1, \ldots \tag{3.14}$$

The construction of the weak form using either energy principles or weighted residual methods will not be detailled here. The reader may easily find it in dozens of textbooks. By substituting (3.14) in the weak form, we obtain the finite element equations of equilibrium in matrix form:

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{D}\dot{\mathbf{U}} + \mathbf{K}(\mathbf{U}) \cdot \mathbf{U} = \mathbf{R}, \tag{3.15}$$

where $\mathbf{M}$ is a constant mass matrix, $\mathbf{D}$ is a constant damping matrix, $\mathbf{K}(\mathbf{U})$ is the stiffness matrix, which is a function of global nodal displacements $\mathbf{U}$, and $\mathbf{R}$ are externally applied loads. This equation is applicable to individual elements as well as the entire element assemblage.

3.3.4 Static system of equations

Under some assumption, (3.15) may be simplified. For instance, let us consider the problem of finding the maximum load that a desk can sustain. It is fair to say that strain and stress within the structure does not fluctuate over time: the system is in an equilibrium state. At all times, the sum of all forces applied on the system is equal to 0. Therefore, we can assume that the equations describing the structure do not depend on time: the problem is said to be *static.* Accelerations can be neglected. In this case, the static system of equations can be easily obtained by merely dropping out the dynamics terms in (3.15). Indeed, the static case may just be seen as a special case of the dynamic equations and the static equations are:

$$\mathbf{K}(\mathbf{U}) \cdot \mathbf{U} = \mathbf{R}. \tag{3.16}$$

3.3.5 A few words on the matrices involved

Mass matrix

The mass matrix $\mathbf{M}$ for the whole domain may be computed by summing the mass contributions from every elements:

$$\mathbf{M} = \sum_e \mathbf{M}^e, \tag{3.17}$$

where $\mathbf{M}^e$ is the mass of element Ω_e given by:

$$\mathbf{M}^e = \int_{V^e} \rho^e \mathbf{H}^T \mathbf{H} dV \tag{3.18}$$

where ρ the mass density, V^e the volume of the element and $\mathbf{H}$ is the shape function matrix defined by (3.10). This integral may be evaluated using a procedure of numerical integration and generally results in a band mass matrix (non-zero entries are confined to a diagonal band, comprising the main diagonal and more diagonals on either side). This method for computing the mass matrix is called *variational mass lumping.*

However, it is very common in finite element analysis to use a diagonal mass matrix. The most common way of achieving this is to define the diagonal entries with the sum of the corresponding rows and then set all other values to 0. The physical interpretation of this is lumping all of the mass in the volume surrounding a node at the node itself. A diagonal mass matrix may offer significant computational and storage advantages. First, it is easily inverted, since the inverse of a diagonal matrix is also diagonal. And a diagonal mass matrix can also be stored simply as a vector. Therefore, such a matrix may be built by directly computing the mass of each element and assigning an equal proportion of this to each of the element's nodes. This method is named *direct mass lumping.* The mass m^e of element Ω_e is given by:

$$m^e = \rho^e \int_{V^e} dV, \tag{3.19}$$

which is integrated over the element using a numerical method. Each diagonal component of the mass matrix then receives a contribution of m^e/m.

Damping matrix

In a similar way to the mass matrix, a damping matrix may be defined by summing the damping contributions from every elements:

$$\mathbf{D} = \sum_{e} \mathbf{D}^e, \tag{3.20}$$

where $\mathbf{D}^e$ is the damping of element Ω_e given by:

$$\mathbf{D}^e = \int_{V^e} \kappa^e \mathbf{H}^T \mathbf{H} dV \tag{3.21}$$

where κ^e is a damping parameter. In practice, the components of damping matrices are difficult to determine experimentally. Therefore, it is common to employ the *Rayleigh damping.* The Rayleigh damping assumes that the damping may be written in the form of:

$$\mathbf{D}^e = \alpha \mathbf{M}^e + \beta \mathbf{K}^e, \tag{3.22}$$

where α and β are coefficients defining mass and stiffness proportional damping components. However, for systems with large degrees of freedom, it is difficult to guess meaningful values of α and β. Chowdhury and Dasgupta (2003) have proposed a method to calculate these coefficients. In practice, further assumptions may only lead to include the mass proportional component ($\beta = 0$), which yields a diagonal damping matrix if a diagonal mass matrix is used.

Stiffness matrix

The expression of the stiffness matrix depends on the chosen strain tensor as well as the appropriate constitutive model for the material. Its general form cannot be explicited. However, it can be demonstrated that $\mathbf{K}^e$ may be written as

$$\mathbf{K}^e = \int_{V^e} \mathbf{B}^T \mathbf{E}\, \mathbf{B}\, dV, \tag{3.23}$$

where $\mathbf{E}$ is a material matrix, which only depends on the constitutive model chosen, and $\mathbf{B}$ is the *strain-displacement matrix* defined as the derivatives of the shape functions (see Liu and Quek (2003); Belytschko et al. (2000) for details).

3.4 Assembly of element equations

We have now derived the equations for each element. However, since the element is physically connected to its neighbours, the resulting algebraic equations will contain more unknowns than the number of algebraic equations. Thus, it becomes necessary to put the elements together to eliminate the extra unknowns. The assembly should be carried out as soon as the element matrices are computed, rather than waiting until the element coefficients of all elements are computed. This would require

Elements	1	2
1	1	2
2	2	3
3	3	4

Table 3.1: Element connectivity information

storage of the coefficients for each element. Instead, we can perform the assembly while we calculate the element matrices.

The assembly operation is actually quite simple. Let us consider an example with stiffness matrices for one-dimensional problem. We assume that we have three two-node elements. And we suppose the element connectivity information given by Table 3.1, which provides the global node number for each local node number within an element.

TODO: add figure with global stiffness matrix and a few arrows issued from the element stiffness matrices

3.5 Solution of global problem

The complete and global system of equations has now been established. Let us recall the general system of equations in dynamic introduced in section 3.3.3, which is also valid for the whole system:

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{D}\dot{\mathbf{U}} + \mathbf{K}(\mathbf{U}) \cdot \mathbf{U} = \mathbf{R}. \tag{3.24}$$

In this section, we will examine the techniques commonly used for solving these equations. In the general case of a dynamic system of equations, the equations are not yet algebraic and still depends on derivatives of the unknown displacements with respect to time. Therefore, an additional step needs to be performed before using a solver by integrating the equations over time. As we will see shortly, the different methods of time integration are classified into two main schemes: *explicit* or *implicit* integration. We begin by introducing the simplest method, the Euler method to explain the concept of an explicit method. Then the central difference method will be detailled. We then describe a few implicit methods, allowing us to highlight the relative advantages of the two types of methods. Finally, the static case will be solved by emphasizing the similarities with implicit systems through the critical step of linearisation of the governing equations (Belytschko et al., 2000).

3.5.1 Explicit time integration

The Euler method

We want to approximate the solution of the following differential equation:

$$y^{'}(t) = f(t, y(t)) \tag{3.25}$$

Note that we restrict ourselves to first-order differential equations (meaning that only the first derivative of y appears in the equation, and no higher derivatives). However, a higher-order equation can easily be converted to a system of first-order equations by introducing extra variables. For example, the second-order equation $y^{''} = -y$ can be rewritten as two first-order equations: $y^{'} = z$ and $z^{'} = -y$. By solving equations like (3.25), we will therefore be able to integrate the global system of equations over time.

We start by replacing the derivative $y^{'}$ by the finite difference approximation:

$$y^{'}(t) \approx \frac{y(t+\Delta t) - y(t)}{\Delta t}, \tag{3.26}$$

which gives:

$$y(t+\Delta t) \approx y(t) + \Delta t y^{'}(t) \tag{3.27}$$

And substituting the last relation in (3.25) yields:

$$y(t+\Delta t) \approx y(t) + \Delta t f(t, y(t)) \tag{3.28}$$

This formula is usually applied in a recursive scheme. We choose a time step Δt and we construct the sequence of time $t^n = n\Delta t$. Using (3.28), we can approximate the exact solution $y(t^n)$ at each time step by y^n:

$$y^{n+1} = y^n + \Delta t f(t^n, y^n). \tag{3.29}$$

This method is called *Euler method* and is said to be explicit because the new value y^{n+1} is defined in terms of variables from the previous time step (y^n).

The central difference method

The central difference method is one of the most popular explicit methods in computational mechanics. This method is based on central difference formulas for the velocity and acceleration. Let us the time t of the simulation be subdivided into time steps Δt. Let us denote by $\mathbf{U}^t$, $\dot{\mathbf{U}}^t$ and $\ddot{\mathbf{U}}^t$ the matrices of nodal displacements, velocities and accelerations, respectively, at time t. The central difference method makes use of the following finite differences approximations for acceleration and velocity:

$$\ddot{\mathbf{U}}^t = \frac{1}{(\Delta t)^2}\left(\mathbf{U}^{t-\Delta t} - 2\mathbf{U}^t + \mathbf{U}^{t+\Delta t}\right) \tag{3.30}$$

$$\dot{\mathbf{U}}^t = \frac{1}{2\Delta t}\left(\mathbf{U}^{t+\Delta t} - \mathbf{U}^{t-\Delta t}\right). \tag{3.31}$$

One of the key advantage of the method is that the stifness term in (3.24) may be computed from

$$\mathbf{K}(\mathbf{U}) \cdot \mathbf{U} = \mathbf{F}(\mathbf{U}) = \sum_e \tilde{\mathbf{F}}^e, \tag{3.32}$$

where $\tilde{\mathbf{F}}^e$ are the global nodal force contributions due to stresses in element Ω_e. $\tilde{\mathbf{F}}$ may be computed using strain-displacement equations, the constitutive equation and the element connectivity. Therefore, let us assume that all nodal forces $\mathbf{F}^t$ at time t are known.

By subtituting (3.30) and (3.31) into (3.24) we obtain:

$$\left(\frac{\mathbf{M}}{(\Delta t)^2}+\frac{\mathbf{D}}{2\Delta t}\right)\mathbf{U}^{t+\Delta t}=\mathbf{R}^t-\mathbf{F}^t+\frac{2\mathbf{M}}{(\Delta t)^2}\mathbf{U}^t+\left(\frac{\mathbf{D}}{2\Delta t}-\frac{\mathbf{M}}{(\Delta t)^2}\right)\mathbf{U}^{t-\Delta t}. \quad (3.33)$$

Both the right-hand side of the equation and the factor in front $\mathbf{U}^{t+\Delta t}$ may be evaluated, which gives the nodal displacements for the next time step. The entire update can be accomplished without solving any system of equations provided that the mass matrix $\mathbf{M}$ and the damping matrix $\mathbf{D}$ are diagonal. As we have seen in section 3.3.5, these approximations are often used in practice. And this is the characteristic of an explicit method: the time integration of the general system of equations for finite element model does not require the solution of any equations. This is a key point because it gives explicit methods a tremendous computational advantage. However, explicit methods are only conditionally stable. If the time step exceeds a critical value, the solution may be erroneous. This critical time step depends on the mesh and the material properties. It will decrease with mesh refinement and increasing stiffness of the material.

3.5.2 Implicit time integration

The backward Euler method

The problem that we wish to solve is the same than the one we introduced with the Euler method, which we reproduce here for convenience:

$$y^{'}(t)=f(t,y(t)) \quad (3.34)$$

This time, instead of using (3.26), we approximate the derivative $y^{'}(t)$ by:

$$y^{'}(t)\approx\frac{y(t)-y(t-\Delta t)}{\Delta t}. \quad (3.35)$$

The update expression for y^n then becomes:

$$y^{n+1}=y^n+\Delta t f(t^{n+1},y^{n+1}). \quad (3.36)$$

The *backward Euler method* is an implicit method, meaning that we have to solve an equation to find y^{n+1}. Of course, it costs time to solve this equation and this cost must be taken into consideration when we select the time integration method to use. The advantage of implicit methods is that they are usually more stable for solving a stiff equation, meaning that larger steps can be used.

The Runge-Kutta method

The accuracy of backward Euler method can be improved by making the solution depends on more values. The 4^{th} order Runge-Kutta method gives the following update equation:

$$y^{n+1} = y^n + \frac{1}{6}\Delta t\left(k_1 + 2k_2 + 2k_3 + k_4\right), \tag{3.37}$$

where :

$$\begin{aligned}
k_1 &= f(t^n, y^n) \\
k_2 &= f(t^n + \frac{1}{2}\Delta t, y^n + \frac{1}{2}\Delta t\, k_1) \\
k_3 &= f(t^n + \frac{1}{2}\Delta t, y^n + \frac{1}{2}\Delta t\, k_2) \\
k_4 &= f(t^n + \Delta t, y^n + \Delta t\, k_3)
\end{aligned}$$

This method is a fourth-order method because the total accumulated error has order Δt^4. It may be seen as a generalisation of the backward Euler method (Press et al., 1992). Again, we have to solve an equation to find y^{n+1}.

3.5.3 Static solutions

First, we recall the general system of equations in static introduced in section 3.3.4, which is also valid for the whole system:

$$\mathbf{K}(\mathbf{U}) \cdot \mathbf{U} = \mathbf{R}. \tag{3.38}$$

The situation is similar to implicit time integration methods for dynamic systems by the fact that we also have to solve a set of algebraic equations to find the nodal displacements. In the general case, the equations are non-linear and therefore quite difficult to solve. A suitable solver for this kind of system is, for instance, the *Newton-Raphson* method. In essence, this method consists of obtaining linear equations instead. This process is called *linearisation*. One may refer to (Belytschko et al., 2000) for more details.

However, if the problem is linear, the stiffness matrix does not depend on the displacement and the resolution is facilitated. The unknown displacements may be computed by:

$$\mathbf{U} = \mathbf{K}^{-1}\,\mathbf{R}. \tag{3.39}$$

Many techniques are available in the literature to solve this type of system, and a few of them are introduced in the next section.

3.5.4 Solvers

We will now examine the techniques commonly used for solving linear systems of algebraic equations. Note that this is only the most classical approaches and the list is obviously not exhaustive. Two categories of techniques are in opposition: *direct* or *iterative* solvers (Press et al., 2002). In any case, we seek the solution for the unknown solution $\mathbf{x}$ of the following system of linear equations:

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{3.40}$$

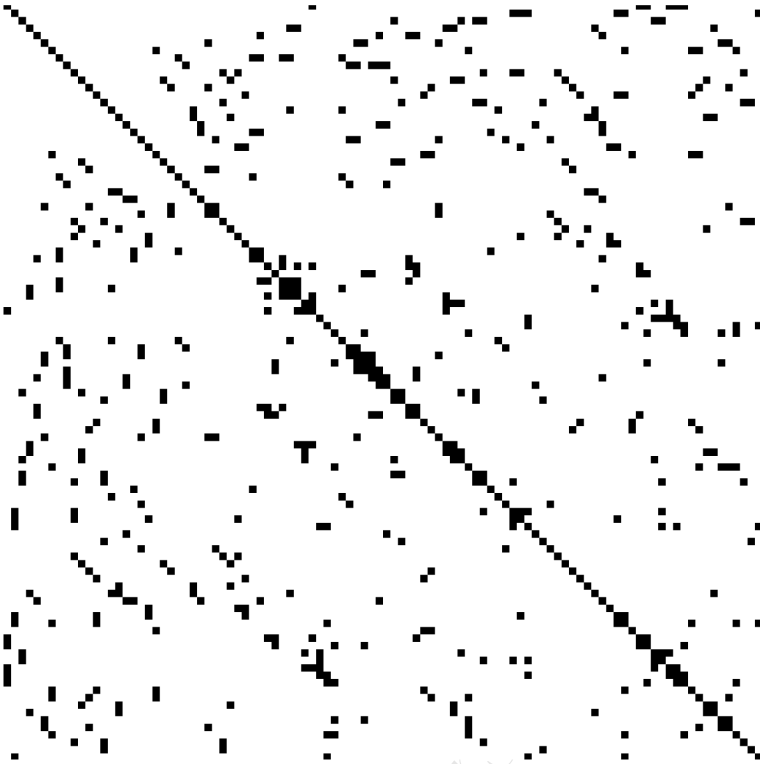

Figure 3.1: Illustration of a sparse matrix for a two-dimensional finite element problem. Each non-zero elements of the matrix is represented in black. Image courtesy of Oleg Alexandrov.

The system may already be linear by nature, or been linearised using techniques such as Newton-Raphson methods. At this point in the discussion, we are therefore only interested in solving a linear system of equations. If the matrix **A** is squared, which is the case in finite element methods, the problem may be written as follows:

$$\mathbf{x} = \mathbf{A}^{-1}\,\mathbf{b}. \tag{3.41}$$

However, only solvers relevant to the context of finite element methods will be discussed. Indeed the matrix **A** takes a specific form in problems solved by finite element methods. The matrix **A** is *sparse*. A sparse matrix is merely a matrix populated mainly with zeros. An example is given by Fig. 3.1 for a finite element problem in two dimensions. Because of the peculiar structure of the matrix involved in finite element computations, methods have been specifically designed to take advantage of this structure. Indeed, it is wasteful to use general methods of linear algebra on such problems, because most of the arithmetic operations devoted to solving the set of equations or inverting the matrix involve zero operands. Furthermore, it is wasteful to reserve storage for zero elements.

Direct solvers

As their name is suggesting, direct solvers make an attempt into finding a solution directly by inverting the matrix $\mathbf{A}$. They execute in a predictable number of operations. We will now introduce a few direct methods in the context of finite element methods.

LU decomposition. The LU decomposition is a matrix decomposition which writes a matrix as the product of a lower triangular matrix and an upper triangular matrix. In other words, $\mathbf{A}$ may be written as:

$$\mathbf{A} = \mathbf{LU}, \tag{3.42}$$

where $\mathbf{L}$ is lower triangular (has elements only on the diagonal and below) and $\mathbf{U}$ is upper triangular (has elements only on the diagonal and above). As an example, for the case of a 4×4 matrix $\mathbf{A}$, we have:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} \alpha_{11} & 0 & 0 & 0 \\ \alpha_{21} & \alpha_{22} & 0 & 0 \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & 0 \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & \alpha_{44} \end{bmatrix} \begin{bmatrix} \beta_{11} & \beta_{12} & \beta_{13} & \beta_{14} \\ 0 & \beta_{22} & \beta_{23} & \beta_{24} \\ 0 & 0 & \beta_{33} & \beta_{34} \\ 0 & 0 & 0 & \beta_{44} \end{bmatrix} \tag{3.43}$$

By subtituting (3.42) into (3.40) we obtain:

$$\mathbf{Ax} = (\mathbf{LU})\mathbf{x} = \mathbf{L}(\mathbf{Ux}) = \mathbf{b}. \tag{3.44}$$

Therefore, we can solve (3.40) by first solving the vector $\mathbf{y}$ such that:

$$\mathbf{Ly} = \mathbf{b} \tag{3.45}$$

and then solving

$$\mathbf{Ux} = \mathbf{y}. \tag{3.46}$$

The advantage of breaking up one linear set of equations into two successive ones is that the solution of a triangular set of equations is quite trivial. Indeed, (3.45) can be solved by:

$$\begin{cases} y_1 = \dfrac{b_1}{\alpha_{11}} \\ y_i = \dfrac{1}{\alpha_{ii}} \left[b_i - \displaystyle\sum_{j=1}^{i-1} \alpha_{ij} y_j \right] \qquad i = 2, 3, \ldots, N. \end{cases} \tag{3.47}$$

for a $N \times N$ matrix $\mathbf{A}$. We then solve (3.46) in a similar way:

$$\begin{cases} x_N = \dfrac{y_N}{\beta_{NN}} \\ x_i = \dfrac{1}{\beta_{ii}} \left[y_i - \displaystyle\sum_{j=i+1}^{N} \beta_{ij} x_j \right] \qquad i = N-1, N-2, \ldots, 1. \end{cases} \tag{3.48}$$

Iterative solvers

An iterative method attempts to solve the system of equations by finding successive approximations to the solution starting from an initial guess. In the case of a system of linear equations, the two main classes of iterative methods are the stationary iterative methods, and the more general Krylov subspace methods. Stationary iterative methods solve a linear system with an operator approximating the original one and based on a measurement of the error in the result (called the residual), form an equation of correction for which this process is repeated. While these methods are simple to derive, convergence is only guaranteed for a limited class of matrices. Examples of stationary iterative methods are the Jacobi method and Gauss?Seidel method. Krylov subspace methods work by forming an orthogonal basis of the sequence of successive matrix powers times the initial residual (the Krylov sequence). The approximations to the solution are then formed by minimising the residual over the subspace formed. The typical method in this class is the conjugate gradient method (CG). In practice, for large system of equations (as often encountered in finite element analysis), direct methods would take too much time and an iterative solvers is commonly used.

The Gauss-Seidel method. For this method, convergence is only guaranteed if the matrix is either diagonally dominant, or symmetric and positive-definite. In a similar way than the LU decomposition, we start by decomposing the matrix $\mathbf{A}$ into a lower triangular component $\mathbf{L}_*$ and a strictly upper triangular component $\mathbf{U}$:

$$\mathbf{A} = \mathbf{L}_* + \mathbf{U}. \tag{3.49}$$

The system of linear equations (3.40) may be rewritten as:

$$\mathbf{L}_*\mathbf{x} = \mathbf{b} - \mathbf{U}\mathbf{x} \tag{3.50}$$

The Gauss-Seidel method is an iterative technique that solves the left-hand side of the expression for $\mathbf{x}$ using previous values for $\mathbf{x}$ on the right-hand side. That is:

$$\mathbf{x}^{n+1} = \mathbf{L}_*^{-1}(\mathbf{b} - \mathbf{U}\mathbf{x}^n). \tag{3.51}$$

Similarly to the LU decomposition method, we take advantage of the triangular form of $\mathbf{L}_*$. The components of $\mathbf{x}^{n+1}$ can be computed sequentially in the following manner:

$$x_i^{n+1} = \frac{1}{a_{ii}}\left[b_i - \sum_{j>i} a_{ij}x_j^n - \sum_{j<i} a_{ij}x_j^{n+1}\right] \quad i = 1, 2, \ldots, N. \tag{3.52}$$

The procedure is usually continued until the changes made by an interation are below a given threshold.

The conjugate gradient method. The simplest class of conjugate gradient method is based on the idea of minimising the quadratic function:

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\,\mathbf{A}\,\mathbf{x} - b^T\,\mathbf{x} + c. \tag{3.53}$$

Indeed, if the matrix $\mathbf{A}$ is symmetric and positive-definite, $f(\mathbf{x})$ is minimised by the solution to $\mathbf{A}\mathbf{x} = \mathbf{b}$(Shewchuk, 1994). The minimisation is carried out by generating a succession of search directions $\mathbf{p}_k$ and improved minimisers $\mathbf{x}_k$. At each stage a quantity α_k is found that minimises $f(\mathbf{x}_k + \alpha_k\mathbf{p}_k)$, and $\mathbf{x}_{k+1}$ is set equal to the new point $\mathbf{x}_k + \alpha_k\mathbf{p}_k$. The $\mathbf{p}_k$ and $\mathbf{x}_k$ are built up in such a way that $\mathbf{x}_{k+1}$ is also the minimizer of f over the whole vector space of directions already taken, that is the space formed by the basis of vector $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_k$. After N iterations, you arrive at the minimiser over the entire vector space, that is, the solution to our linear system of equations.

Later, the *biconjugate gradient method* was introduced as a generalisation of ordinary conjugate gradient to solve systems where the matrix $\mathbf{A}$ was not necessarily symmetric and positive-definite (see Belytschko et al. (2000) for details).

Part II

Solid organs modelling

Chapter 4

STATE OF ART: FEM

4.1 Expose the problem: how to model solid organs?

4.2 Techniques using geometrical kind of approaches

4.2.1 Shape matching

4.3 Techniques relying on physics

4.3.1 Mass-spring

4.4 Techniques based on continuum mechanics

4.4.1 FEM with mesh (highlight evolution: linear, topological changes, co-rotational, non-linear, GPU)

4.4.2 Meshless (supposedly good for handling topological changes but other issues

CHAPTER 5

INTRODUCTION OF TLED

5.1 Description of the algorithm (neo-Hookean, explicit integration, independent elements, parallelisable on GPU)

5.2 Visco-elasticity and anisotropy added (MICCAI 2008; MedIA 2009)

CHAPTER 6

GPU IMPLEMENTATION OF TLED

6.1 What is GPGPU

6.2 Re-formulation of the algorithm for its Cg implementation

6.3 CUDA implementation/optimisations (ISBMS 2008a)

6.4 Implementation into SOFA

6.4.1 Presentation of SOFA project and architecture

6.4.2 TLED in SOFA and release in open-source

Part III

Hollow organs modelling

CHAPTER 7

STATE OF ART: HOLLOW STRUCTURES

7.1 Expose the problem: models presented for solid organs are insufficient. How to model hollow organs?

7.1.1 Colonoscopy simulator project: needs for colon

7.1.2 Cataract surgery, stenosis: other needs for implants/blood vessels

7.2 Non-physic approaches (computer graphics stuff)

7.3 Physically accurate approches (plates/shells)

CHAPTER 8

MASS-SPRING MODEL FOR COLON IMPLEMENTED ON GPU (ISBMS 2008b)

8.1 Colon is big, we need a fast model

8.2 Limitations (difficult to link it to actual physical properties)

CHAPTER 9

A CO-ROTATIONAL TRIANGULAR SHELL MODEL (ISBMS 2010)

9.1 Model description

9.2 Validation

9.3 Application to implant deployment simulation in cataract surgery

CHAPTER 10

’SHELL MESHING’ TECHNIQUE (MICCAI 2010)

10.1 State of art: reconstruction/simplification

10.2 Our method

Part IV

Conclusion

CHAPTER 11

SUMMARY

Olivier Comas - Sat 7^{th} of August 2010 at 17:15

CHAPTER 12

DISCUSSION AND PERSPECTIVES (INTERACTION SOLID/HOLLOW ORGANS)

APPENDIX A

TENSORS

Appendix B

The weighted residual method

The *method of weighted residuals* is an approximation technique for solving differential equations. A subclass of this method, the *Galerkin method of weighted residuals*, is often used to derive the element equations for the finite element method.

Let us suppose that we have a linear differential operator D acting on a function u to produce a function p:

$$D(u(x)) = p(x). \tag{B.1}$$

We wish to approximate u by a function $\tilde{u}$, which is a linear combination of basis functions chosen from a linearly independent set, that is:

$$u \approx \tilde{u} = \sum_{i=1}^{n} a_i \phi_i. \tag{B.2}$$

Now, when substituted into the differential operator D, the result of the operation is not, in general, $p(x)$. Hence, an error or *residual* will exist:

$$R(x) = D(\tilde{u}(x) - p(x)) \neq 0. \tag{B.3}$$

The idea behind the weighted residual method is to force the residual to zero in some average sense over the domain:

$$\int_X R(x) W_i(x) dx = 0, \quad i = 1, 2, \ldots, n \tag{B.4}$$

where the number of *weight functions* W_i is exactly equal to the number of unknown constants a_i in $\tilde{u}$. The result is a set of n algebraic equations for the unknown constants a_i. Depending on the choice for W_i, several sub-methods can be derived: collocation, sub-domain, least squares, Ritz, Galerkin etc. For instance, the weight function used by the Galerkin method TODO: need to check this: Wi are the shape functions? is:

$$W_i = \frac{\partial \tilde{u}}{\partial a_i}. \tag{B.5}$$

References

[Belytschko et al., 2000] T. Belytschko, W. K. Liu and B. Moran. *Nonlinear Finite Elements for Continua and Structures.* Wiley, 1 ed., 2000.

[Biswas et al., 1976] D. Biswas, K. S. Ram and S. S. Rao. *Application of 'natural coordinate system' in the finite element solution of multigroup neutron diffusion equation.* Annals of Nuclear Energy, vol. 3, no. 11-12, pages 465–469, 1976.

[Bonnet and Wood, 1997] J. Bonnet and R. D. Wood. *Nonlinear Continuum Mechanics For Finite Element Analysis.* Cambridge University Press, 1997.

[Chowdhury and Dasgupta, 2003] I. Chowdhury and S. Dasgupta. *Computation of Rayleigh damping coefficients for large systems.* The Electronic Journal of Geotechnical Engineering, vol. 8, no. C, 2003.

[Chui et al., 2007] C. Chui, E. Kobayashi, X. Chen, T. Hisada and I. Sakuma. *Transversely isotropic properties of porcine liver tissue: experiments and constitutive modelling.* Medical and Biological Engineering and Computing, vol. 45, no. 1, pages 99–106, 2007.

[Gent, 2001] A. N. Gent. *Engineering With Rubber: How to Design Rubber Components.* Hanser Gardner Publications, 2001.

[Lai et al., 1996] W. Lai, D. Rubin and E. Krempl. *Introduction to Continuum Mechanics.* Butterworth-Heinemann, third ed., 1996.

[Lautrup, 2005] B. Lautrup. *Physics of continuous matter: exotic and everyday phenomena in the macroscopic world.* Institute of Physics Publishing, 2005.

[Liu and Quek, 2003] G. Liu and S. S. Quek. *Finite Element Method: A Practical Course.* Butterworth-Heinemann, 2003.

[MacDonald, 2007] B. J. MacDonald. *Practical Stress Analysis with Finite Elements.* Glasnevin Publishing, 1 ed., 2007.

[Melvin et al., 1973] J. Melvin, R. Stalnaker and V. Roberts. *Impact injury mechanisms in abdominal organs.* SAE Transactions, , no. 730968, pages 115–126, 1973.

[Miller and Chinzei, 1997] K. Miller and K. Chinzei. *Constitutive modelling of brain tissue: Experiment and theory.* Journal of Biomechanics, vol. 30, no. 11-12, pages 1115 – 1121, 1997.

[Press et al., 1992] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling. *Numerical Recipes in FORTRAN 77: The Art of Scientific Computing.* Cambridge University Press, 2 ed., 1992.

[Press et al., 2002] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery. *Numerical Recipes in C++: The Art of Scientific Computing.* Cambridge University Press, 2 ed., 2002.

[Reddy, 1993] J. N. Reddy. *Introduction to the Finite Element Method.* McGraw-Hill, 2 ed., 1993.

[Reddy, 2007] J. N. Reddy. *An Introduction to Continuum Mechanics.* Cambridge University Press, 2007.

[Sedef et al., 2006] M. Sedef, E. Samur and C. Basdogan. *Real-Time Finite-Element Simulation of Linear Viscoelastic Tissue Behavior Based on Experimental Data.* IEEE Comput. Graph. Appl., vol. 26, no. 6, pages 58–68, 2006, ISSN 0272-1716.

[Shewchuk, 1994] J. R. Shewchuk. *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain.* Tech. rep., Shool of Computer Science - Carnegie Mellon University, 1994.

[Spencer, 1980] A. Spencer. *Continuum Mechanics.* Longman Group Limited, 1980.