Version 1.0

Theory Manual

Written and Edited by

Steve Maas and Jeff Weiss

Musculoskeletal Research Laboratories
Department of Bioengineering, and
Scientific Computing and Imaging Institute
University of Utah

72 S. Central Campus Drive, Room 2646
Salt Lake City, Utah

steve.maas@utah.edu

jeff.weiss@utah.edu

Last Updated: August 29, 2007

Table of Contents

Chapter 1. Introduction

1.1. Overview of FEBio

FEBio is an implicit, nonlinear finite element solver that is specifically designed for applications in biomechanics. It offers modeling scenarios, constitutive models and boundary conditions that are relevant for this particular field. This section describes the available features of FEBio. All these features can be used together seamlessly, giving the user a powerful tool for solving 3D computational problems in biomechanics.

FEBio supports two analysis types, namely *quasi-static* and *quasi-static poroelastic*. In a *quasi-static* analysis the (quasi-) static response of the system is sought; inertial terms are ignored. In a *quasi-static poroelastic* analysis a coupled solid-fluid problem is solved. The latter analysis type is useful for modeling tissues that have high water content and the explicit modeling of fluid movement relative to the solid phase is important.

Several nonlinear constitutive models are available which allows the user to model the often complicated biological tissue behavior. Several isotropic constitutive models are supported such as Neo-Hookean, Mooney-Rivlin and Veronda-Westmann. These models have a nonlinear stress-strain response. A linear elastic model is also available for small strain scenarios and validation. In addition to the isotropic models there are several transversely isotropic models available. These materials show anisotropic behavior in a single preferred direction and are useful for modeling biological tissues such as tendons, muscles and other tissues that contain fibers. FEBio also contains a *rigid body* material model, which can be used to model rigid structures whose deformation is negligible compared to the deformable geometry.

Biological tissues can interact in very complicated ways. Therefore FEBio supports a wide range of boundary conditions to model these interactions. These include prescribed displacements, nodal forces, and pressure forces. Deformable models can also be connected to rigid bodies so that the user can also model prescribed rotations and torques. Rigid bodies can also be connected with joints. Even more complicated interactions can be modeled using sliding interfaces. A sliding surface is defined between two surfaces that are allowed to separate and slide across each other but are not allowed to penetrate. In addition the user may specify a body force which can be used to model the effects of, for instance, gravity or base acceleration.

1.2. About this document

This document is a part of a set of three manuals that accompany FEBio: the User's Manual, describing how to use FEBio, the Developer's Manual for users who which to modify or add features to the code, and the Theory Manual (this manual), which describes the theory behind most of the FEBio algorithms.

The purpose of this manual is to provide theoretical background on many of the algorithms that are implemented in FEBio. In this way the user can develop a better understanding of how the program works and how it can be used to create well defined biomechanical simulations. The authors have tried to be as detailed as possible to make the text coherent and comprehensible, but due to the complexity of some of the topics, some descriptions only skim the surface. Many of the theoretical ideas discussed in this manual can and have filled entire bookshelves. The explanations contained herein should be sufficient to give the reader a basic understanding of the theoretical developments. References to textbooks and the primary literature are provided for further reading.

Chapter 2 starts with a brief overview of some of the important concepts in continuum mechanics. Readers who are already familiar with this field can skip this chapter, although the material may be useful to get familiar with the notation and terminology used in this manual.

Chapter 3 describes the nonlinear finite element method. It also explains the Newton-Raphson method, which is the basis for most implementations of the nonlinear finite element method. A more specialized version of this algorithm, the BFGS method, is described as well since it is used in FEBio.

In Chapter 4 the different element models that are available in FEBio are described in detail. FEBio currently supports 3D solid elements, such as the linear hexahedral, pentahedral and tetrahedral elements, as well as shell elements, such as a quadrilateral and triangular element. Note that currently the shell elements can only be used with rigid materials.

Chapter 5 contains a detailed description of the material models in FEBio. Most of these models are based on hyperelasticity which is introduced in chapter 2. Several transversely isotropic materials are described as well. In this chapter we also discuss the biphasic material and its implementation in FEBio.

Chapter 6 describes the basics of the theory of contact and coupling. In FEBio the user can connect the different parts of the geometry in a variety of ways. There are rigid interfaces where a deformable model is attached to a rigid model, rigid joints where two or more rigid bodies connect, and sliding interfaces where two surfaces are allowed to separate and slide across each other but are not allowed to penetrate. The various contact and coupling algorithms are discussed as well together with their implementation in FEBio.

Chapter 2. Continuum Mechanics

This chapter contains an overview of some of the important concepts from continuum mechanics and establishes some of the notation and terminology that will be used in the rest of this document. The section begins by introducing the important concepts of deformation, stress and strain. Next the concept of hyperelasticity is discussed. Finally the concept of virtual work is discussed. This concept will be used later to derive the nonlinear finite element equations.

2.1. Vectors and Tensors

It is assumed that the reader is familiar with the concepts of vectors and tensors. This section summarizes the notation and some useful relations that will be used throughout the manual.

Vectors are denoted by small, bold letters, e.g. **v**. Their components will be denoted by v_i, where, unless otherwise stated, Latin under scripts such as *i* or *I* will range from 1 to 3. In matrix form a vector will be represented as a column vector and its transpose as a row vector:

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}, \quad \mathbf{v}^T = \begin{pmatrix} v_1, & v_2, & v_3 \end{pmatrix}. \tag{2.1}$$

The following products are defined between vectors. Assume **u**, **v** are vectors. Also note that we use the Einstein summation convention throughout this manual.

The *dot* or *scalar product*:

$$\mathbf{u} \cdot \mathbf{v} = u_i v_j . \tag{2.2}$$

The *vector* or *cross product*:

$$\mathbf{u} \times \mathbf{v} = \begin{bmatrix} u_2 v_3 - u_3 v_2 \\ u_3 v_1 - u_1 v_3 \\ u_1 v_2 - u_2 v_1 \end{bmatrix}. \tag{2.3}$$

The *outer product*:

$$\left(\mathbf{u} \otimes \mathbf{v}\right)_{ij} = u_i v_j . \tag{2.4}$$

Note that vectors are also known as first order tensors. Scalars are known as zero order tensors.

Second order tensors are denoted by bold, capital letters, e.g. **A**. Some exceptions will be made to remain consistent with the current literature. For instance, the Cauchy stress tensor is denoted by $\boldsymbol{\sigma}$. However, the nature of the objects will always be clear from the context. The following operations on tensors are defined. Assume **A** and **B** are second-order tensors.

The *double contraction*, or *tensor inner product* is defined as:

$$\mathbf{A} : \mathbf{B} = A_{ij} B_{ij} . \tag{2.5}$$

The *trace* is defined as:

$$\operatorname{tr} \mathbf{A} = A_{ii} . \tag{2.6}$$

In general the components of tensors will change under a change of coordinate system. Nevertheless, certain intrinsic quantities associated with them will remain invariant under such a transformation. The scalar product between two vectors is such an example. The double contraction between two second-order tensors is another example. The following set of invariants for second-order tensors is commonly used.

$$\begin{aligned} I_1 &= \operatorname{tr} \mathbf{A} \\ I_2 &= \frac{1}{2}\left(\left(\operatorname{tr} \mathbf{A} \right)^2 - \operatorname{tr} \mathbf{A}^2 \right) \\ I_3 &= \det \mathbf{A} \end{aligned} \tag{2.7}$$

Higher order tensors will be denoted by bold, capital, curly symbols, e.g. $\boldsymbol{\mathcal{A}}$. An example of a third-order tensor is the *permutation tensor* $\boldsymbol{\mathcal{E}}$, whose components are 1 for an even permutation of $(1,2,3)$, -1 for an odd permutation of $(1,2,3)$ and zero otherwise. An example of a fourth order tensor is the elasticity tensor $\boldsymbol{\mathcal{C}}$ which, in linear elasticity theory, relates the small strain tensor $\boldsymbol{\varepsilon}$ and the Cauchy stress tensor $\boldsymbol{\sigma} = \boldsymbol{\mathcal{C}} : \boldsymbol{\varepsilon}$.

In the implementation of the FE method it is often convenient to write symmetric second-order tensors using Voigt notation. In this notation the components of a 2nd order tensor **A** will be represented and stored as a column vector:

$$\left[\underline{\mathbf{A}} \right] = \begin{bmatrix} A_{11} \\ A_{22} \\ A_{33} \\ A_{12} \\ A_{23} \\ A_{13} \end{bmatrix} . \tag{2.8}$$

2.2. The Directional Derivative

In later sections the nonlinear finite element method will be formulated. Anticipating an iterative solution method, it will be necessary to linearize the quantities involved. This linearization process will utilize a construction called the *directional derivative* [2] which we shall introduce briefly here.

The directional derivative of a function $f(\mathbf{x})$ is defined as follows.

$$Df(\mathbf{x})[\mathbf{u}] = \left. \frac{d}{d\varepsilon} \right|_{\varepsilon=0} f(\mathbf{x} + \varepsilon \mathbf{u}) \tag{2.9}$$

The quantity **x** may be a scalar, a vector or even a vector of unknown functions. For instance, consider a scalar function $f(\mathbf{x})$, where **x** is the position vector in $\mathbb{R}^3$. In this case the directional derivative is given by,

$$\begin{aligned} Df(\mathbf{x}) &= \left.\frac{d}{d\varepsilon}\right|_{\varepsilon=0} f(\mathbf{x}+\varepsilon\mathbf{u}) \\ &= \frac{\partial f}{\partial x_i}u_i \\ &= \nabla f \cdot \mathbf{u} \end{aligned} \tag{2.10}$$

Here, the symbol ∇ ("del") depicts the familiar gradient operator.

The linearization of a function implies that it is approximated by a linear function. Using the directional derivative a function *f* can be linearized as follows,

$$f(\mathbf{x}+\mathbf{u}) \cong f(\mathbf{x}) + Df(\mathbf{x})[\mathbf{u}]. \tag{2.11}$$

2.3. Deformation, Strain and Stress

2.3.1. The deformation gradient tensor

Consider the deformation of an object when it moves from the initial or *reference configuration* to the *current configuration*. The location of the material particles in the reference configuration are denoted by **X** and are known as the *material coordinates.* Their location in the current configuration is denoted by **x** and known as the *spatial coordinates*. The *deformation map* φ, which is a mapping from $\mathbb{R}^3$ to $\mathbb{R}^3$, maps the coordinates of a material point to the spatial configuration:

$$\mathbf{x} = \varphi(\mathbf{X}). \tag{2.12}$$

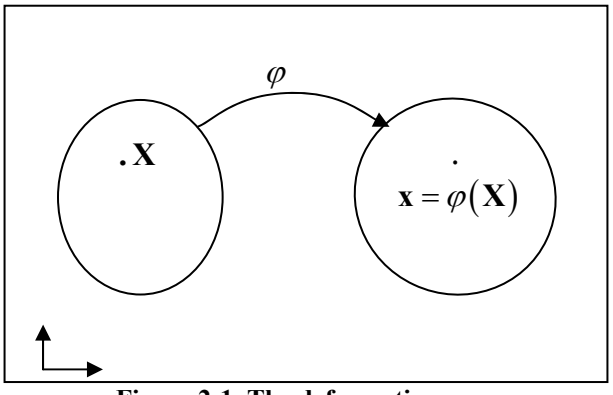

Figure 2-1. The deformation map

The displacement map **u** is defined as the difference between the spatial and material coordinates:

$$\mathbf{x} = \mathbf{X} + \mathbf{u}(\mathbf{X}). \tag{2.13}$$

The *deformation gradient tensor* is defined as

$$\mathbf{F} = \frac{\partial \varphi}{\partial \mathbf{X}}. \tag{2.14}$$

This tensor relates an infinitesimal vector in the reference configuration $d\mathbf{X}$ to the corresponding vector in the current configuration:

$$d\mathbf{x} = \mathbf{F} \cdot d\mathbf{X}. \tag{2.15}$$

The determinant of the deformation tensor $J = \det \mathbf{F}$ gives the volume change, or equivalently the change in density:

$$\rho_0 = \rho J. \tag{2.16}$$

Here ρ_0 is the density in the reference configuration and ρ is the current density.

When dealing with incompressible and nearly incompressible materials it will prove useful to separate the volumetric and the deviatoric (distortional) components of the deformation gradient. Such a separation must ensure that the deviatoric component, namely $\tilde{\mathbf{F}}$, does not produce any change in volume. Noting that the determinant of the deformation gradient gives the volume ratio, the determinant of $\tilde{\mathbf{F}}$ must therefore satisfy,

$$\det \tilde{\mathbf{F}} = 1. \tag{2.17}$$

This condition can be achieved by choosing $\tilde{\mathbf{F}}$ as,

$$\tilde{\mathbf{F}} = J^{-1/3}\mathbf{F}. \tag{2.18}$$

2.3.2. Strain

The *right Cauchy-Green deformation tensor* is defined as follows:

$$\mathbf{C} = \mathbf{F}^T\mathbf{F}. \tag{2.19}$$

This tensor is an example of a *material tensor* and is a function of the material coordinates **X**. The *left Cauchy-Green deformation tensor* is defined as follows:

$$\mathbf{B} = \mathbf{F}\mathbf{F}^T. \tag{2.20}$$

This tensor is an example of a *spatial tensor* and is a function of the spatial coordinates **x**. The implementation of the updated Lagrangian finite element method used by FEBio is described in the spatial configuration.

The left and right deformation tensors can also be split into volumetric and deviatoric components. With the use of (2.18), the deviatoric deformation tensors are:

$$\begin{aligned} \tilde{\mathbf{C}} &= \tilde{\mathbf{F}}^T\tilde{\mathbf{F}} = J^{-2/3}\mathbf{C} \\ \tilde{\mathbf{B}} &= \tilde{\mathbf{F}}\tilde{\mathbf{F}}^T = J^{-2/3}\mathbf{B} \end{aligned}. \tag{2.21}$$

The deformation tensors defined above are not good candidates for strain measures since in the absence of strain they become the unity tensor $\mathbf{I}$ $\left((\mathbf{I})_{ij} = \delta_{ij}\right)$. However, they can be used to define strain measures. The *Green-Lagrange strain tensor* is defined as:

$$\mathbf{E} = \frac{1}{2}\left(\mathbf{C} - \mathbf{I}\right). \tag{2.22}$$

This tensor is a material tensor. Its spatial equivalent is known as the *Almansi strain tensor* and is defined as:

$$\mathbf{e} = \frac{1}{2}\left(\mathbf{I} - \mathbf{B}^{-1}\right). \tag{2.23}$$

In the limit of small displacement gradients, the components of both strain tensors are identical, resulting in the *small strain tensor* or *infinitesimal strain tensor*:

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left(\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \left(\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\right)^T\right). \tag{2.24}$$

Note that the small strain tensor is also the linearization of the Green Lagrange strain,

$$D\mathbf{E}[\mathbf{u}] = \mathbf{F}^T\boldsymbol{\varepsilon}\mathbf{F} \tag{2.25}$$

2.3.3. Stress

The traction **t** on a plane bisecting the body is given by,

$$\mathbf{t} = \boldsymbol{\sigma}\cdot\mathbf{n}, \tag{2.26}$$

where $\boldsymbol{\sigma}$ is the *Cauchy stress tensor* and **n** is the outer unit normal vector to the plane. It can be shown that by the conservation of angular momentum that this tensor is symmetric ($\sigma_{ij} = \sigma_{ji}$) [1]. The Cauchy stress tensor, a spatial tensor, is the actual physical stress, that is, the force per unit deformed area. To simplify the equations of continuum mechanics, especially when working in the material configuration, several other stress measures are available. The *Kirchhoff stress tensor* is defined as

$$\boldsymbol{\tau} = J\boldsymbol{\sigma}. \tag{2.27}$$

The *first Piola-Kirchhoff stress tensor* is given as

$$\mathbf{P} = J\boldsymbol{\sigma}\mathbf{F}^{-T}. \tag{2.28}$$

Note that **P**, like **F**, is unsymmetric. Also, like **F**, **P** is known as a *two-point* tensor, meaning it is neither a material nor a spatial tensor. Since we have two strain tensors, one spatial and one material tensor, it would be nice to have similar stress measures. The Cauchy stress is a spatial tensor and the *second Piola-Kirchhoff (2nd PK) stress tensor*, given as

$$\mathbf{S} = J\mathbf{F}^{-1}\boldsymbol{\sigma}\mathbf{F}^{-T}, \tag{2.29}$$

is a material tensor. The inverse relations are:

$$\boldsymbol{\sigma} = \frac{1}{J}\boldsymbol{\tau}, \quad \boldsymbol{\sigma} = \frac{1}{J}\mathbf{P}\mathbf{F}^T, \quad \boldsymbol{\sigma} = \frac{1}{J}\mathbf{F}\mathbf{S}\mathbf{F}^T. \tag{2.30}$$

In many practical applications it is physically relevant to separate the hydrostatic stress and the deviatoric stress $\tilde{\boldsymbol{\sigma}}$ of the Cauchy stress tensor.

$$\boldsymbol{\sigma} = \tilde{\boldsymbol{\sigma}} + p\mathbf{I} \tag{2.31}$$

Here, the pressure is defined as $p = \frac{1}{3}\operatorname{tr}\boldsymbol{\sigma}$. Note that the deviatoric Cauchy stress tensor satisfies $\operatorname{tr}\tilde{\boldsymbol{\sigma}} = 0$.

In the linearization of the finite element equations the directional derivative of the 2nd PK stress tensor needs to be calculated. Using the chain rule, a linear relationship between the directional derivative of **S** and the linearized strain $D\mathbf{E}[\mathbf{u}]$ can be obtained.

$$D\mathbf{S}[\mathbf{u}] = \mathbb{C} : D\mathbf{E}[\mathbf{u}] \tag{2.32}$$

Here, $\mathbb{C}$ is a fourth-order tensor known as the *material elasticity tensor*. Its components are given by,

$$\mathcal{C}_{IJKL} = \frac{\partial S_{IJ}}{\partial E_{KL}} = \frac{4\partial^2 \Psi}{\partial C_{IJ} \partial C_{KL}} \tag{2.33}$$

The spatial equivalent – the *spatial elasticity tensor* – can be obtained by,

$$\mathcal{c}_{ijkl} = \frac{1}{J} F_{iI} F_{jJ} F_{kK} F_{lL} \mathcal{C}_{IJKL} \,. \tag{2.34}$$

2.4. Hyperelasticity

When the constitutive behavior is only a function of the current state of deformation, the material is *elastic*. In the special case when the work done by the stresses during a deformation is only dependent on the initial state and the final state, the material is termed *hyperelastic* and its behavior is path-independent. As a consequence of the path-independence a *strain energy function* per unit undeformed volume can be defined as the work done by the stresses from the initial to the final configuration:

$$\Psi\left(\mathbf{F}(\mathbf{X}),\mathbf{X}\right) = \int_{t_0}^{t} \mathbf{P}\left(\mathbf{F}(\mathbf{X}),\mathbf{X}\right) : \dot{\mathbf{F}}\, dt \,. \tag{2.35}$$

The rate of change of the potential is then given by,

$$\dot{\Psi}\left(\mathbf{F}(\mathbf{X}),\mathbf{X}\right) = \mathbf{P} : \dot{\mathbf{F}} \tag{2.36}$$

Or alternatively,

$$P_{iJ} = \sum_{i,J=1}^{3} \frac{\partial \Psi}{\partial F_{iJ}} \dot{F}_{iJ} \tag{2.37}$$

Comparing (2.36) with (2.37) reveals that

$$\mathbf{P}\left(\mathbf{F}(\mathbf{X}),\mathbf{X}\right) = \frac{\partial \Psi\left(\mathbf{F}(\mathbf{X}),\mathbf{X}\right)}{\partial \mathbf{F}} \,. \tag{2.38}$$

This general constitutive equation can be further developed by observing that, as a consequence of the objectivity requirement, Ψ may only depend on **F** through the stretch tensor **U** and must be independent on the rotation component **R**. For convenience, however, Ψ is often expressed as a function of $\mathbf{C} = \mathbf{U}^2 = \mathbf{F}^T\mathbf{F}$. Noting that $\frac{1}{2}\dot{\mathbf{C}} = \dot{\mathbf{E}}$ is work conjugate to the second Piola-Kirchhoff stress **S**, establishes the following general relationships for hyperelastic materials:

$$\dot{\Psi} = \frac{\partial \Psi}{\partial \mathbf{C}} : \dot{\mathbf{C}} = \frac{1}{2}\mathbf{S} : \dot{\mathbf{C}}, \quad \boxed{\mathbf{S}\left(\mathbf{C}(\mathbf{X}),\mathbf{X}\right) = 2\frac{\partial \Psi}{\partial \mathbf{C}} = \frac{\partial \Psi}{\partial \mathbf{E}}} \,. \tag{2.39}$$

2.4.1. Isotropic Hyperelasticity

The hyperelastic constitutive equations discussed so far are unrestricted in their application. Isotropic material symmetry is defined by requiring the constitutive behavior to be independent of the material axis chosen and, consequently, Ψ must only be a function of the invariants of **C**,

$$\Psi\left(\mathbf{C}(\mathbf{X}),\mathbf{X}\right) = \Psi\left(I_1, I_2, I_3, \mathbf{X}\right) \tag{2.40}$$

where the invariants of **C** are defined here as,

$$I_1 = \text{tr}\,\mathbf{C} = \mathbf{C} : \mathbf{I}$$
$$I_2 = \frac{1}{2}\left[\left(\text{tr}\,\mathbf{C}\right)^2 - \text{tr}\,\mathbf{C}^2\right] \tag{2.41}$$
$$I_3 = \det \mathbf{C} = J^2$$

As a result of the isotropic restriction, the second Piola-Kirchhoff stress tensor can be written as,

$$\mathbf{S} = 2\frac{\partial \Psi}{\partial \mathbf{C}} = 2\frac{\partial \Psi}{\partial I_1}\frac{\partial I_1}{\partial \mathbf{C}} + 2\frac{\partial \Psi}{\partial I_2}\frac{\partial I_2}{\partial \mathbf{C}} + 2\frac{\partial \Psi}{\partial I_3}\frac{\partial I_3}{\partial \mathbf{C}} \tag{2.42}$$

The second order tensors formed by the derivatives of the invariants with respect to **C** can be evaluated as follows,

$$\frac{\partial I_1}{\partial \mathbf{C}} = \mathbf{I}$$
$$\frac{\partial I_2}{\partial \mathbf{C}} = I_1\mathbf{I} - \mathbf{C} \tag{2.43}$$
$$\frac{\partial I_3}{\partial \mathbf{C}} = I_3\mathbf{C}^{-1}$$

Introducing expressions (2.43) into equation (2.42) enables the second Piola-Kirchhoff stress to be evaluated as,

$$\mathbf{S} = 2\left\{\left(\Psi_1 + I_1\Psi_2 + I_2\Psi_3\right)\right)\mathbf{I} - \left(\Psi_2 + I_1\Psi_3\right)\mathbf{C}\right\} + \Psi_3\mathbf{C}^2 \tag{2.44}$$

where $\Psi_I = \partial\Psi / \partial I_1$, $\Psi_2 = \partial\Psi / \partial I_2$, and $\Psi_3 = \partial\Psi / \partial I_3$.

The Cauchy stresses can now be obtained indirectly from the second Piola-Kirchhoff stresses by using (2.30).

$$\boldsymbol{\sigma} = 2\left\{\left(\Psi_1 + I_1\Psi_2 + I_2\Psi_3\right)\right)\mathbf{B} - \left(\Psi_2 + I_1\Psi_3\right)\mathbf{B}^2\right\} + \Psi_3\mathbf{B}^3 \tag{2.45}$$

Note that in this equation Ψ_1, Ψ_2, and Ψ_3 still involve derivatives with respect to the invariants of **C**. However, since the invariants of **B** are identical to those of **C**, the quantities Ψ_1, Ψ_2 and Ψ_3 may also be considered to be the derivatives with respect to the invariants of **B**.

2.4.2. Nearly-Incompressible Hyperelasticity

A material is considered incompressible if it shows no change in volume, or otherwise stated if $J = 1$ holds throughout the entire body. It can be shown [2] that if the material is incompressible the hyperelastic constitutive equation becomes,

$$\mathbf{S} = 2\frac{\partial \tilde{\Psi}}{\partial \mathbf{C}} + pJ\mathbf{C}^{-1} \tag{2.46}$$

where $\tilde{\Psi} = \Psi\left(\tilde{\mathbf{C}}\right)$ and p is the hydrostatic pressure. The presence of J may seem unnecessary, but retaining J has the advantage that equation (2.46) remains valid in the nearly incompressible case. Further, in practical terms, a finite element analysis rarely enforces $J = 1$ in a pointwise manner, and hence its retention may be important for the

evaluation of stresses. The process of defining constitutive equations in the case of nearly incompressible hyperelasticity is simplified by adding a volumetric energy component $U(J)$ to the distortional component $\tilde{\Psi}(\mathbf{C})$,

$$\Psi(\mathbf{C}) = \tilde{\Psi}(\mathbf{C}) + U(J) \tag{2.47}$$

The second Piola-Kirchhoff tensor for a material defined by (2.47) is obtained in the standard manner with the help of equation (2.42).

$$\begin{aligned} \mathbf{S} &= 2\frac{\partial \Psi}{\partial \mathbf{C}} \\ &= 2\frac{\partial \tilde{\Psi}}{\partial \mathbf{C}} + 2\frac{dU}{dJ}\frac{\partial J}{\partial C} \\ &= 2\frac{\partial \tilde{\Psi}}{\partial \mathbf{C}} + pJ\mathbf{C}^{-1} \end{aligned} \tag{2.48}$$

where the pressure is defined as

$$p = \frac{dU}{dJ}. \tag{2.49}$$

An example for U that will be used later in the definition of the constitutive models is

$$U(J) = \frac{1}{2}\kappa(\ln J)^2. \tag{2.50}$$

The parameter κ will be used later as a penalty factor that will enforce the (nearly-) incompressible constraint. However, κ can represent a true material coefficient, namely the bulk modulus, for a compressible material that happens to have a hyperelastic strain energy function in the form of (2.47). In the case where the dilatational energy is given by (2.50) the pressure is

$$p = \kappa\frac{\ln J}{J}. \tag{2.51}$$

2.4.3. Transversely Isotropic Hyperelasticity

Transverse isotropy can be introduced by adding a vector field representing the material preferred direction explicitly into the strain energy [3]. We require that the strain energy depends on a unit vector field $\mathbf{a}^0$, which described the local fiber direction in the undeformed configuration. When the material undergoes deformation, the vector $\mathbf{a}^0(\mathbf{X})$ may be described by a unit vector field $\mathbf{a}(\varphi(\mathbf{X}))$. In general, the fibers will also undergo length change. The fiber stretch, λ, can be determined in terms of the deformation gradient and the fiber direction in the undeformed configuration,

$$\lambda\mathbf{a} = \mathbf{F}\cdot\mathbf{a}^0. \tag{2.52}$$

Also, since $\mathbf{a}$ is a unit vector,

$$\lambda^2 = \mathbf{a}^0\cdot\mathbf{C}\cdot\mathbf{a}^0. \tag{2.53}$$

The strain energy function for a transversely isotropic material, $\Psi(\mathbf{C},\mathbf{a}^0,\mathbf{X})$ is an isotropic function of $\mathbf{C}$ and $\mathbf{a}^0\otimes\mathbf{a}^0$. It can be shown [1] that the following set of invariants are sufficient to describe the material fully:

$$I_1 = \operatorname{tr}\mathbf{C}, \quad I_2 = \frac{1}{2}\left[\left(\operatorname{tr}\mathbf{C}\right)^2 - \operatorname{tr}\mathbf{C}^2\right], \quad I_3 = \det C = J^2 , \tag{2.54}$$

$$I_4 = \mathbf{a}^0 \cdot \mathbf{C} \cdot \mathbf{a}^0, \quad I_5 = \mathbf{a}^0 \cdot \mathbf{C}^2 \cdot \mathbf{a}^0 . \tag{2.55}$$

The strain energy function can be written in terms of these invariants such that,

$$\Psi\left(\mathbf{C}, \mathbf{a}^0, \mathbf{X}\right) = \Psi\left(I_1\left(\mathbf{C}\right), I_2\left(\mathbf{C}\right), I_3\left(\mathbf{C}\right), I_4\left(\mathbf{C}, \mathbf{a}^0\right), I_5\left(\mathbf{C}, \mathbf{a}^0\right)\right). \tag{2.56}$$

The second Piola-Kirchhoff can now be obtained in the standard manner:

$$\mathbf{S} = 2\frac{\partial \Psi}{\partial \mathbf{C}} = 2\sum_{i=1}^{5} \frac{\partial \Psi}{\partial I_i}\frac{\partial I_i}{\partial \mathbf{C}}. \tag{2.57}$$

In the transversely isotropic constitutive models described in Chapter 5 it is further assumed that the strain energy function can be split into the following terms,

$$\Psi\left(\mathbf{C}, \mathbf{a}^0\right) = \Psi_1\left(I_1, I_2, I_3\right) + \Psi_2\left(I_4\right) + \Psi_3\left(I_1, I_2, I_3 I_4\right). \tag{2.58}$$

The strain energy function Ψ_1 represents the material response of the isotropic ground substance matrix, Ψ_2 represents the contribution from the fiber family (e.g. collagen), and Ψ_3 is the contribution from interactions between the fibers and matrix. The form (2.58) generalizes many constitutive equations that have been successfully used in the past to describe biological soft tissues e.g. [4-6]. While this relation represents a large simplification when compared to the general case, it also embodies almost all of the material behavior that one would expect from transversely isotropic, large deformation matrix-fiber composites.

Chapter 3. The Nonlinear FE Method

This chapter discusses the basic principles of the nonlinear finite element method. The chapter begins with a short introduction to the weak formulation and the principle of virtual work. Next the important concept of linearization is discussed and applied to the principle of virtual work. Finally the Newton-Raphson procedure and its application to the non-linear finite element method is described.

3.1. Weak formulation

Generally, the finite element formulation is established in terms of a weak form of the differential equations under consideration. In the context of solid mechanics this implies the use of the virtual work equation:

$$\delta W = \int_v \boldsymbol{\sigma} : \delta \mathbf{d} dv - \int_v \mathbf{f} \cdot \delta \mathbf{v} dv - \int_{\partial v} \mathbf{t} \cdot \delta \mathbf{v} da = 0 . \tag{3.1}$$

Here, $\delta \mathbf{v}$ is a virtual velocity and $\delta \mathbf{d}$ is the virtual rate of deformation tensor. This equation is known as the *spatial virtual work equation* since it is formulated using spatial quantities only. We can also define the *material virtual work equation* by expressing the principle of virtual work using only material quantities.

$$\delta W = \int_V \mathbf{S} : \delta \dot{\mathbf{E}} dV - \int_V \mathbf{f}_0 \cdot \delta \mathbf{v} dV - \int_{\partial V} \mathbf{t}_0 \cdot \delta \mathbf{v} dA = 0 \tag{3.2}$$

Here, $\mathbf{f}_0 = J\mathbf{f}$ is the body force per unit undeformed volume and $\mathbf{t}_0 = \mathbf{t}\left(da/dA\right)$ is the traction vector per unit initial area.

3.1.1. Linearization

Equation (3.1) is the starting point for the nonlinear finite element method. It is highly nonlinear and any method attempting to solve this equation, such as the Newton-Raphson method, necessarily has to be iterative.

To linearize the finite element equations, we need to calculate the directional derivative, introduced in section 2.2, to the principle of virtual work in equation (3.1). In an iterative procedure the quantity ϕ will be approximated by a trial solution ϕ_k. The virtual work equation linearized around this trial solution gives

$$\delta W\left(\phi_k, \delta \mathbf{v}\right) + D\delta W\left(\phi_k, \delta \mathbf{v}\right)\left[\mathbf{u}\right] = 0 . \tag{3.3}$$

The directional derivative of the virtual work will eventually lead to the definition of the stiffness matrix. In order to proceed it is convenient to split the virtual work into an internal and external virtual work component.

$$D\delta W\left(\phi, \delta \mathbf{v}\right)\left[\mathbf{u}\right] = D\delta W_{\text{int}}\left(\phi, \delta \mathbf{v}\right)\left[\mathbf{u}\right] - D\delta W_{\text{ext}}\left(\phi, \delta \mathbf{v}\right)\left[\mathbf{u}\right], \tag{3.4}$$

where,

$$\delta W_{\text{int}}\left(\phi, \delta \mathbf{v}\right) = \int_v \boldsymbol{\sigma} : \delta \mathbf{d} dv , \tag{3.5}$$

and

$$\delta W_{\text{ext}}\left(\varphi,\delta\mathbf{v}\right)=\int_{v}\mathbf{f}\cdot\delta\mathbf{v}dv+\int_{\partial v}\mathbf{t}\cdot\delta\mathbf{v}da\,. \tag{3.6}$$

The result is listed here without details of the derivation – see [2] for details. The linearization of the internal virtual work is given by

$$D\delta W_{\text{int}}\left(\phi,\delta\mathbf{v}\right)\left[\mathbf{u}\right]=\int_{v}\delta\mathbf{d}:\mathbb{C}:\varepsilon dv+\int_{v}\boldsymbol{\sigma}:\left[\left(\nabla\mathbf{u}\right)^{T}\nabla\delta\mathbf{v}\right]dv\,. \tag{3.7}$$

Notice that this equation is symmetric in $\delta\mathbf{v}$ and $\mathbf{u}$. This symmetry will, upon discretization, yield a symmetric tangent matrix.

The external virtual work has contributions from both body forces and surface tractions. The precise form of the linearized external virtual work depends on the form of these forces. FEBio currently supports gravity as a body force, $\mathbf{f}=\rho\mathbf{g}$. Since this force is independent of the geometry the contribution to the linearized external work is zero. For surface tractions, normal pressure forces may be represented in FEBio. The linearized external work for this type of traction is given by

$$\begin{aligned}D\delta W_{\text{ext}}^{p}\left(\phi,\delta\mathbf{v}\right)\left[\mathbf{u}\right]=&\frac{1}{2}\int_{A_{\xi}}p\frac{\partial\mathbf{x}}{\partial\xi}\cdot\left[\left(\frac{\partial\mathbf{u}}{\partial\eta}\times\delta\mathbf{v}\right)+\left(\frac{\partial\delta\mathbf{v}}{\partial\eta}\right)\right]d\xi d\eta\\&-\frac{1}{2}\int_{A_{\xi}}p\frac{\partial\mathbf{x}}{\partial\eta}\cdot\left[\left(\frac{\partial\mathbf{u}}{\partial\xi}\times\delta\mathbf{v}\right)+\left(\frac{\partial\delta\mathbf{v}}{\partial\xi}\right)\right]d\xi d\eta\end{aligned}\,. \tag{3.8}$$

Discretization of this equation will also lead to a symmetric component of the tangent matrix.

3.1.2. Discretization

The basis of the finite element method is that the domain of the problem (that is the volume of the object under consideration) is divided into smaller subunits, called *finite elements*. In the case of *isoparametric elements* it is further assumed that each element has a local coordinate system, named the *natural coordinates*, and the coordinates and shape of the element are discretized using the same functions. The discretization process is established by interpolating the geometry in terms of the coordinates $\mathbf{X}_a$ of the *nodes* that define the geometry of a finite element, and the *shape functions*.

$$\mathbf{X}=\sum_{a=1}^{n}N_a\left(\xi_1,\xi_2,\xi_3\right)\mathbf{X}_a\,, \tag{3.9}$$

where *n* is the number of nodes and ξ_i are the natural coordinates. Similarly the motion is described in terms of the current position $\mathbf{x}_a\left(t\right)$ of the *same* particles:

$$\mathbf{x}\left(t\right)=\sum_{a=1}^{n}N_a\mathbf{x}_a\left(t\right). \tag{3.10}$$

Quantities such as displacement, velocity and virtual velocity can be discretized in a similar way.

In deriving the discretized equilibrium equations, the integrations performed over the entire volume can be written as a sum of integrations constrained to the volume of an element. For this reason the discretized equations are defined in terms of integrations

over a particular element *e*. The discretized equilibrium equations for this particular element per node is given by

$$\delta W^{(e)}\left(\phi, N_a \delta \mathbf{v}\right) = \delta \mathbf{v}_a \cdot \left(\mathbf{T}_a^{(e)} - \mathbf{F}_a^{(e)}\right), \tag{3.11}$$

where

$$\begin{aligned} T_a^{(e)} &= \int_{v^{(e)}} \sigma \nabla N_a dv \\ F_a^{(e)} &= \int_{v^{(e)}} N_a \mathbf{f} dv + \int_{\partial v^{(e)}} N_a \mathbf{t} da \end{aligned}. \tag{3.12}$$

The linearization of the internal virtual work can be split into a "material" and an "initial stress" component [2]:

$$\begin{aligned} D\delta W_{\text{int}}^{(e)}\left(\phi, \delta \mathbf{v}\right)\left[\mathbf{u}\right] &= \int_{v^{(e)}} \delta \mathbf{d} : \mathbb{C} : \varepsilon dv + \int_{v^{(e)}} \boldsymbol{\sigma} : \left[\left(\nabla \mathbf{u}\right)^T \nabla \delta \mathbf{v}\right] dv \\ &= D\delta W_c^{(e)}\left(\phi, \delta \mathbf{v}\right)\left[\mathbf{u}\right] + D\delta W_\sigma^{(e)}\left(\phi, \delta \mathbf{v}\right)\left[\mathbf{u}\right] \end{aligned}. \tag{3.13}$$

The constitutive component can be discretized as follows:

$$D\delta W_c^{(e)}\left(\phi, \delta \mathbf{v}\right)\left[\mathbf{u}\right] = \delta \mathbf{v}_a \cdot \left(\int_{v^{(e)}} \mathbf{B}_a^T \mathbf{D} \mathbf{B}_b dv\right) \mathbf{u}_b . \tag{3.14}$$

The term in parentheses defines the constitutive component of the tangent matrix relating node *a* to node *b* en element *e*:

$$\mathbf{K}_{c,ab}^{(e)} = \int_{v^{(e)}} \mathbf{B}_a^T \mathbf{D} \mathbf{B}_b dv . \tag{3.15}$$

Here, the linear strain-displacement matrix **B** relates the displacements to the small-strain tensor in Voigt Notation

$$\underline{\boldsymbol{\varepsilon}} = \sum_{a=1}^{n} \mathbf{B}_a \mathbf{u}_a . \tag{3.16}$$

Or written out completely,

$$\mathbf{B}_a = \begin{bmatrix} \partial N_a / \partial x & 0 & 0 \\ 0 & \partial N_a / \partial y & 0 \\ 0 & 0 & \partial N_a / \partial z \\ \partial N_a / \partial y & \partial N_a / \partial x & 0 \\ 0 & \partial N_a / \partial z & \partial N_a / \partial y \\ \partial N_a / \partial z & 0 & \partial N_a / \partial z \end{bmatrix}. \tag{3.17}$$

The spatial constitutive matrix **D** is constructed from the components of the fourth-order tensor $\mathbb{C}$ using the following table; $\mathbf{D}_{IJ} = \mathbb{C}_{ijkl}$ where

I/J	**i/k**	**j/l**
1	1	1
2	2	2
3	3	3

4	1	2
5	2	3
6	1	3

The initial stress component can be written as follows:

$$D\delta W_{\sigma}^{(e)}\left(\phi, N_a\delta\mathbf{v}\right)\left[N_b\mathbf{u}_b\right] = \int_{v^{(e)}}\left(\nabla N_a\cdot\boldsymbol{\sigma}\nabla N_b\right)\mathbf{I}dv\,. \tag{3.18}$$

For the pressure component of the external virtual work we find

$$D\delta W_{p}^{(e)}\left(\phi, N_a\delta\mathbf{v}_a\right)\left[N_b\mathbf{u}_b\right] = \delta\mathbf{v}_a\cdot\mathbf{K}_{p,ab}^{(e)}\mathbf{u}_b\,, \tag{3.19}$$

where,

$$\mathbf{K}_{p,ab}^{(e)} = \boldsymbol{\mathcal{E}}\mathbf{k}_{p,ab}^{(e)},\quad \mathbf{k}_{p,ab}^{(e)} = \frac{1}{2}\int_{A_\xi} p\frac{\partial\mathbf{x}}{\partial\xi}\left(\frac{\partial N_a}{\partial\eta}N_b - \frac{\partial N_b}{\partial\eta}N_a\right)d\xi d\eta\,. \tag{3.20}$$
$$+\frac{1}{2}\int_{A_\xi} p\frac{\partial\mathbf{x}}{\partial\eta}\left(\frac{\partial N_a}{\partial\xi}N_b - \frac{\partial N_b}{\partial\xi}N_a\right)d\xi d\eta$$

3.2. Newton-Raphson method

The Newton-Raphson method (also known as "Newton's method", "Full Newton method" or "the Newton method") is the basis for solving the nonlinear finite element equations. It comes in several variants of which we will describe the *Full Newton method* and the *BFGS method*. The latter variation is important since it provides several advantages over the full Newton method and it is this method that is implemented in FEBio [7].

3.2.1. Full Newton Method

In the previous section the discretized equilibrium equations and the discretized linearized virtual work were determined. Consequently the Newton-Raphson equation (3.3) can be formulated in its discretized version as follows,

$$\delta\mathbf{v}^T\cdot\mathbf{K}\cdot\mathbf{u} = -\delta\mathbf{v}^T\cdot\mathbf{R} \tag{3.21}$$

Since the virtual velocities $\delta\mathbf{v}$ are arbitrary, a discretized Newton-Raphson scheme can be formulated as follows,

$$\mathbf{K}\left(\mathbf{x}_k\right)\cdot\mathbf{u} = -\mathbf{R}\left(\mathbf{x}_k\right);\quad \mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{u} \tag{3.22}$$

This is the basis of the Newton-Raphson method. For each iteration, both the stiffness matrix and the residual vector are re-evaluated and an increment **u** is calculated by pre-multiplying both side of the above equation by $\mathbf{K}^{-1}$. This procedure is repeated until some convergence criteria are satisfied.

The formation of the stiffness matrix and, especially, calculation of its inverse, are computationally expensive. Variations of the Newton-Raphson method that do not require the reevaluation of the stiffness matrix for every iteration have been developed. In quasi-Newton methods, instead of reevaluating the stiffness matrix for each iteration, a quick update is calculated. One particular method that has been quite successful in the

field of computational solid mechanics is the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method [7], which is described in the next section.

3.2.2. BFGS Method

As stated above the BFGS method belongs to a class of methods called quasi-Newton methods. These methods involve updating the stiffness matrix (or rather its inverse) to provide an approximation to the exact matrix. A displacement increment is defined as

$$\mathbf{d}_k = \mathbf{x}_k - \mathbf{x}_{k-1}, \tag{3.23}$$

and an increment in the residual is defined as

$$\mathbf{G}_k = \mathbf{R}_{k-1} - \mathbf{R}_k. \tag{3.24}$$

The updated matrix $\mathbf{K}_k$ should satisfy the quasi-Newton equation:

$$\mathbf{K}_k\mathbf{d}_k = \mathbf{G}_k. \tag{3.25}$$

In order to calculate this update we first calculate a displacement increment:

$$\mathbf{u} = \mathbf{K}_{k-1}^{-1}\mathbf{R}_{k-1}. \tag{3.26}$$

This displacement vector defines a “direction” for the actual displacement increment. A line search (see next section) can now be applied to determine the optimal displacement increment:

$$\mathbf{x}_k = \mathbf{x}_{k-1} + s\mathbf{u}, \tag{3.27}$$

where s is determined from the line search. With the updated position calculated we can now evaluate $\mathbf{R}_k$ and also, using equations (3.23) and (3.24) $\mathbf{d}_k$ and $\mathbf{G}_k$. We can now move on to the evaluation of the stiffness update. In the BFGS method this update can be expressed as follows:

$$\mathbf{K}_k^{-1} = \mathbf{A}_k^T\mathbf{K}_{k-1}^{-1}\mathbf{A}_k, \tag{3.28}$$

where the matrix **A** is an $n \times n$ matrix of the simple form:

$$\mathbf{A}_k = \mathbf{I} + \mathbf{v}_k\mathbf{w}_k^T. \tag{3.29}$$

The vectors **v** and **w** are given by

$$\mathbf{v}_k = -\left(\frac{\mathbf{d}_k^T\mathbf{G}_k}{\mathbf{d}_k^T\mathbf{K}_{k-1}\mathbf{d}_k}\right)^{1/2}\mathbf{K}_{k-1}\mathbf{d}_k - \mathbf{G}_k, \tag{3.30}$$

$$\mathbf{w}_k = \frac{\mathbf{d}_k}{\mathbf{d}_k^T\mathbf{G}_k}. \tag{3.31}$$

The vector $\mathbf{K}_{k-1}\mathbf{d}_k$ is equal to $s\mathbf{R}_{k-1}$ and has already been calculated.

To avoid numerically dangerous updates, the condition number c of the updating matrix **A** is calculated:

$$c = \left(\frac{\mathbf{d}_k^T\mathbf{G}_k}{\mathbf{d}_k^T\mathbf{K}_{k-1}^{-1}\mathbf{d}_k}\right)^{1/2}. \tag{3.32}$$

When this number exceeds a preset tolerance the update is not performed.

Considering the actual computation involved, it should be noted that using the matrix updates defined above, the calculation of the search direction in (3.26) can be rewritten as,

$$\mathbf{u} = \left(\mathbf{I} + \mathbf{w}_{k-1}\mathbf{v}_{k-1}^T\right)\ldots\left(\mathbf{I} + \mathbf{w}_1\mathbf{v}_1^T\right)\mathbf{K}^{-1}\left(\mathbf{I} + \mathbf{v}_1\mathbf{w}_1^T\right)\cdots\left(\mathbf{I} + \mathbf{v}_{k-1}\mathbf{w}_{k-1}^T\right)\mathbf{R}_{k-1} \tag{3.33}$$

Hence, the search direction can be computed without explicitly calculating the updated matrices or performing any additional costly matrix factorizations as required in the full Newton-Raphson method.

3.2.3. Line Search Method

A powerful technique often used to improve the convergence rate of Newton based methods is the *line search method*. In this method the displacement vector **u** is considered as an optimal search direction, but allowing the magnitude to be controlled by a parameter *s*.

$$\mathbf{x}_{k+1} = \mathbf{x}_k + s\mathbf{u} \tag{3.34}$$

The value of *s* is usually chosen so that the total potential energy $W(s) = W(\mathbf{x}_k + s\mathbf{u})$ at the end of the iteration is minimized in the direction of **u**. This is equivalent to the requirement that the residual force $\mathbf{R}(\mathbf{x}_k + s\mathbf{u})$ at the end of the iteration is orthogonal to **u**.

$$R(s) = \mathbf{u}^T\mathbf{R}(\mathbf{x}_k + s\mathbf{u}) = 0 \tag{3.35}$$

However, in practice it is sufficient to obtain a value of *s* such that,

$$\left|R(s)\right| < \rho\left|R(0)\right| \tag{3.36}$$

where typically a value of $\rho = 0.5$ is used. Under normal conditions the value $s = 1$ automatically satisfies equation (3.36) and therefore few extra operations are involved. However, when this is not the case a more suitable value for *s* needs to be obtained. For this reason it is convenient to approximate $R(s)$ as a quadratic in *s*.

$$R(s) \approx (1-s)R(0) + R(1)s^2 = 0 \tag{3.37}$$

which yields a value for *s* as,

$$s = \frac{r}{2} \pm \sqrt{\left(\frac{r}{2}\right)^2 - r}, \quad r = \frac{R(0)}{R(1)} \tag{3.38}$$

If $r < 0$, the square root is positive and a first improved value for *s* is obtained.

$$s_1 = \frac{r}{2} + \sqrt{\left(\frac{r}{2}\right)^2 - r} \tag{3.39}$$

If $r > 0$ the *s* can be obtained by using the value that minimizes the quadratic function, that is, $s_1 = r/2$. This procedure is now repeated with $R(1)$ replaced by $R(s_1)$ until equation (3.36) is satisfied.

Chapter 4. Element Library

FEBio provides several element types for finite element discretization. This chapter describes these elements in more detail.

4.1. Solid Elements

The 3D solid elements available in FEBio are *isoparametric elements*. All of the solid elements are formulated in a global Cartesian coordinate system. For all these elements, a local coordinate system is defined as well, the so-called isoparametric coordinates. The global position vector **x** can be written as a function of the isoparametric coordinates in the following sense:

$$\mathbf{x}(r,s,t) = \sum_{i=1}^{n} N_i(r,s,t)\mathbf{x}_i . \tag{4.1}$$

Here, *n* is the number of nodes, *r*, *s* and *t* are the isoparametric coordinates, N_i are the element shape functions and $\mathbf{x}_i$ are the spatial coordinates of the element nodes. The same parametric interpolation is used for interpolation of other scalar and vector quantities.

All elements in FEBio are integrated numerically. This implies that integrals over the volume of the element are approximated by a sum:

$$\begin{aligned} \int_{v^e} f(\mathbf{x})\,dv &= \int_{\square^e} f(\mathbf{r})\,J(\mathbf{r})\,d\square \\ &\cong \sum_{i=1}^{m} f(\mathbf{r}_i)\,J_i w_i \end{aligned} . \tag{4.2}$$

Here, *m* is the number of integration points , $\mathbf{r}_i$ are the location of the integration points in isoparametric coordinates, *J* is the Jacobian of the transformation $\mathbf{x} = \mathbf{x}(r,s,t)$, and w_i is a weight associated with the integration point.

Most fully integrated solid elements are unsuitable for the analysis of (nearly-) incompressible material behavior. To deal with this type of materials a three-field element implementation is available in FEBio [8].

4.1.1. Hexahedral elements

FEBio implements an 8-node trilinear hexahedral element. This element is also known as a *brick* element. The shape functions for these elements are defined in function of the isoparametric coordinates *r*, *s* and *t* and are given below.

$$
\begin{aligned}
N_1 &= \frac{1}{}(1-r)(1-s)(1-t)\\
N_2 &= \frac{1}{2}(1+r)(1-s)(1-t)\\
N_3 &= \frac{1}{2}(1+r)(1+s)(1-t)\\
N_4 &= \frac{1}{2}(1-r)(1+s)(1-t)\\
N_5 &= \frac{1}{2}(1-r)(1-s)(1+t)\\
N_6 &= \frac{1}{2}(1+r)(1-s)(1+t)\\
N_7 &= \frac{1}{2}(1+r)(1+s)(1+t)\\
N_8 &= \frac{1}{2}(1-r)(1+s)(1+t)
\end{aligned}
\tag{4.3}
$$

4.1.2. Pentahedral Elements

Pentahedral elements (also knows as “wedge” elements) consist of six nodes and six faces. Their shape functions are defined in function of the isoparametric coordinates *r*, *s* and *t* and are given as follows.

$$
\begin{aligned}
N_1 &= \frac{1}{2}(1-r-s)(1-t)\\
N_2 &= \frac{1}{2}r(1-t)\\
N_3 &= \frac{1}{2}s(1-t)\\
N_4 &= \frac{1}{2}(1-r-s)(1+t)\\
N_5 &= \frac{1}{2}r(1+t)\\
N_6 &= \frac{1}{2}s(1+t)
\end{aligned}
\tag{4.4}
$$

4.1.3. Tetrahedral Elements

Linear 4-node tetrahedral elements are also available in FEBio. Their shape functions are defined in function of the isoparametric coordinates *r*, *s* and *t*.

$$\begin{aligned} N_1 &= 1 - r - s - t \\ N_2 &= r \\ N_3 &= s \\ N_4 &= t \end{aligned} \tag{4.5}$$

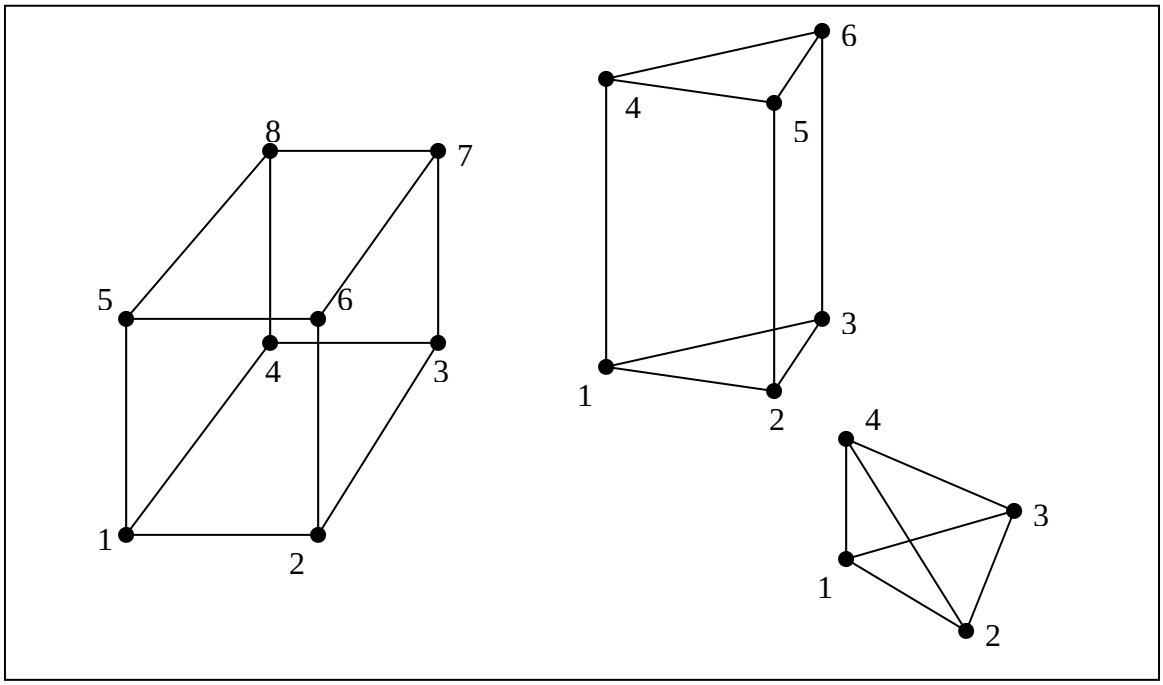

Figure 4-1. Different solid element types that are available in FEBio.

4.2. Shell Elements

Currently FEBio does not support deformable shell elements. It only supports a *rigid shell* model, where the shell element has to belong to a rigid body. In other words, rigid bodies can be fully described using a surface model alone. The surface model can be composed of quadrilateral or triangular elements.

For surface elements only two isoparametric coordinates are required, *s* and *t*.

$$\mathbf{x}(r,s) = \sum_{i=1}^{n} N_i(r,s)\mathbf{x}_i \tag{4.6}$$

4.2.1. Quadrilateral shells

For quadrilateral shells, the shape functions are given by,

$$
\begin{aligned}
N_1 &= \frac{1}{4}(1-r)(1-s) \\
N_2 &= \frac{1}{4}(1+r)(1-s) \\
N_3 &= \frac{1}{4}(1+r)(1+s) \\
N_4 &= \frac{1}{4}(1-r)(1+s)
\end{aligned}
\tag{4.7}
$$

4.2.2. Triangular shells

For triangular shell elements, the shape functions are given by,

$$
\begin{aligned}
N_1 &= 1-r-s \\
N_2 &= r \\
N_3 &= s
\end{aligned}
\tag{4.8}
$$

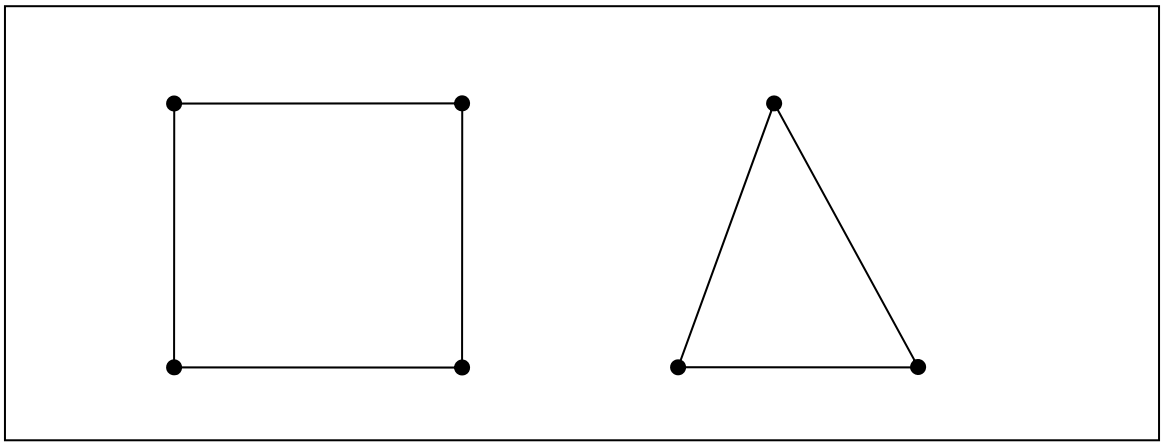

Figure 4-2. Different rigid shell elements available in FEBio

Chapter 5. Constitutive Models

This chapter describes the theoretical background behind the constitutive models that are available in FEBio. Most materials are derived from a hyperelastic strain-energy function. Please consult section 2.4 for more background information on this type of materials.

5.1. Linear Elasticity

In the theory of linear elasticity the Cauchy stress tensor is a linear function of the small strain tensor $\boldsymbol{\varepsilon}$:

$$\boldsymbol{\sigma} = \boldsymbol{\mathcal{C}} : \boldsymbol{\varepsilon} . \tag{5.1}$$

Here $\boldsymbol{\mathcal{C}}$ is the fourth-order elasticity tensor that contains the material properties. In the most general case this tensor has 21 independent parameters. However, in the presence of material symmetry the number of independent parameters is greatly reduced. For example, in the case of isotropic linear elasticity only two independent parameters remain.

For linear elasticity, the elasticity tensor is given by

$$\mathcal{C}_{ijkl} = \lambda \delta_{ij} \delta_{kl} + 2\mu \delta_{ik} \delta_{jl} . \tag{5.2}$$

Using this relationship we can write the stress-strain relationship as follows:

$$\sigma_{ij} = \lambda \left(\operatorname{tr} \boldsymbol{\varepsilon} \right)^2 \delta_{ij} + 2\mu \varepsilon_{ij} . \tag{5.3}$$

If we represent the stress and strain by Voigt vectors, we can rewrite this in matrix form:

$$\begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{23} \\ \sigma_{13} \end{pmatrix} = \begin{pmatrix} \lambda+2\mu & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda+2\mu & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{pmatrix} \begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \lambda_{12} \\ \lambda_{23} \\ \lambda_{13} \end{pmatrix} . \tag{5.4}$$

The strain measures λ_{ij} are the *engineering strains* and are given by $\lambda_{ij} = 2\varepsilon_{ij}$. The material coefficients λ and μ are known as the Lamé parameters. They relate to the more familiar Young's modulus *E* and Poisson's ratio ν as follows:

$$\begin{aligned} \lambda &= \frac{\nu E}{(1+\nu)(1-2\nu)}, \\ \mu &= \frac{E}{2(1+\nu)}. \end{aligned} \tag{5.5}$$

It is often convenient to express the material properties with the bulk modulus *K* and shear modulus *G*.

$$K = \lambda + \frac{2}{3}\mu = \frac{E}{3(1-2\nu)}$$
$$G = \mu = \frac{E}{2(1+\nu)} \tag{5.6}$$

The theoretical range of the Poisson's ratio for an isotropic material is $-1 \le \nu \le 0.5$. Materials with Poisson's ratio (close to) 0.5 are known as (nearly-) incompressible materials. For these materials the bulk modulus approaches infinity. Most materials have a positive Poisson's ratio although there do exist some materials with a negative ratio. These materials are known as *auxetic* materials and they have the unique property that they expand under tension.

The linear stress-strain relationship can also be derived from a strain-energy function such as in the case of hyperelastic materials. In this case the linear strain-energy is given by,

$$W = \frac{1}{2}\boldsymbol{\varepsilon} : \mathbf{C}\boldsymbol{\varepsilon} \tag{5.7}$$

The stress is then similarly derived from $\boldsymbol{\sigma} = \frac{\partial W}{\partial \boldsymbol{\varepsilon}}$. In the case of isotropic elasticity we can simplify (5.7),

$$W = \frac{1}{2}\lambda\left(\operatorname{tr}\boldsymbol{\varepsilon}\right)^2 + \mu\boldsymbol{\varepsilon} : \boldsymbol{\varepsilon}\,. \tag{5.8}$$

The Cauchy stress is now given by

$$\boldsymbol{\sigma} = \lambda\left(\operatorname{tr}\boldsymbol{\varepsilon}\right)\mathbf{I} + 2\mu\boldsymbol{\varepsilon}\,. \tag{5.9}$$

5.2. St. Venant-Kirchhoff elasticity

The linear elastic material model as described in section 5.1 is only valid for small strains and small rotations. A first modification to this model to the range of nonlinear deformations is given by the St. Venant-Kirchhoff model. Although this model is still only useful for small strains it does allow for large rotations. For the isotropic case it can be derived from the following hyperelastic strain-energy function:

$$W = \frac{1}{2}\lambda\left(\operatorname{tr}\mathbf{E}\right)^2 + \mu\mathbf{E}:\mathbf{E}\,. \tag{5.10}$$

The second Piola-Kirchhoff stress can be derived from this:

$$\mathbf{S} = \frac{\partial W}{\partial \mathbf{E}} = \lambda\left(\operatorname{tr}\mathbf{E}\right)\mathbf{I} + 2\mu\mathbf{E}\,. \tag{5.11}$$

Note that these equations are similar to the corresponding equations in the linear elastic case, only the small strain tensor is replaced by the Lagrangian elasticity tensor **E**.

5.3. Neo-Hookean Hyperelasticity

This is a compressible Neo-Hookean material. It is derived from the following hyperelastic strain energy function [2]:

$$W = \frac{\mu}{2}\left(I_1 - 3\right) - \mu\ln J + \frac{\lambda}{2}\left(\ln J\right)^2.$$

The parameters μ and λ are the Lamé parameters from linear elasticity. For small strains and rotations this model reduces to the isotropic linear elastic model.

The Neo-Hookean material is an extension of Hooke's law for the case of large deformations. It is useable for plastics and rubber-like substances. A generalization of this model is the Mooney-Rivlin material which is often used to describe the elastic response of biological tissue.

In FEBio this constitutive model uses a standard displacement-based element formulation, so care must be taken when modeling materials with nearly-incompressible material behavior to avoid element locking.

5.4. Mooney-Rivlin Hyperelasticity

This material model is a hyperelastic Mooney-Rivlin type with uncoupled deviatoric and volumetric behavior. The uncoupled strain energy W is given by:

$$W = C_1\left(\tilde{I}_1 - 3\right) + C_2\left(\tilde{I}_2 - 3\right) + \frac{1}{2}K\left(\ln J\right)^2.$$

C_1 and C_2 are the Mooney-Rivlin material coefficients, $\tilde{I}_1$ and $\tilde{I}_2$ are the invariants of the deviatoric part of the right Cauchy-Green deformation tensor, $\tilde{\boldsymbol{C}} = \tilde{\boldsymbol{F}}^T\tilde{\boldsymbol{F}}$, where $\tilde{\boldsymbol{F}} = J^{(-1/3)}\boldsymbol{F}$, $\boldsymbol{F}$ is the deformation gradient and $J = \det\left(\boldsymbol{F}\right)$ is the Jacobian of the deformation. When $C_2 = 0$, this model reduces to an uncoupled version of the

incompressible neo-Hookean constitutive model. This material model uses a three-field element formulation, interpolating displacements as linear field variables and pressure and volume ratio as piecewise constant on each element [8].

5.5. Veronda-Westmann Hyperelasticity

This model is similar to the Mooney-Rivlin model in that it also uses an uncoupled deviatoric dilatational strain energy. However, in this case the strain energy is given by an exponential form:

$$W = C_1\left[e^{\left(C_2\left(\tilde{I}_1-3\right)\right)}-1\right]-\frac{C_1C_2}{2}\left(\tilde{I}_2-3\right)+U\left(J\right).$$

The dilatational term U is identical to the Mooney-Rivlin model.

This material model was the result from the research of the elastic response of skin tissues [9].

5.6. Transversely Isotropic Hyperelastic

This constitutive model can be used to represent a material that has a single preferred fiber direction and was developed for application to biological soft tissues [3, 10, 11]. It can be used to model tissues such as tendons, ligaments and muscle. The elastic response of the tissue is assumed to arise from the resistance of the fiber family and an isotropic matrix. It is assumed that the strain energy function can be written as follows:

$$W = F_1\left(\tilde{I}_1,\tilde{I}_2\right)+F_2\left(\tilde{\lambda}\right)+\frac{K}{2}\left[\ln\left(J\right)\right]^2.$$

Here $\tilde{I}_1$ and $\tilde{I}_2$ are the first and second invariants of the deviatoric version of the right Cauchy Green deformation tensor $\tilde{\boldsymbol{C}}$ and $\tilde{\lambda}$ is the deviatoric part of the stretch along the fiber direction ($\tilde{\lambda}^2 = \boldsymbol{a}_0\cdot\tilde{\boldsymbol{C}}\cdot\boldsymbol{a}_0$, where $\boldsymbol{a}_0$ is the initial fiber direction), and $J=\det\left(\boldsymbol{F}\right)$ is the Jacobian of the deformation (volume ratio). The function F_1 represents the material response of the isotropic ground substance matrix and is the same as the Mooney-Rivlin form specified above, while F_2 represents the contribution from the fiber family. The strain energy of the fiber family is as follows:

$$\tilde{\lambda}\frac{\partial F_2}{\partial\tilde{\lambda}} = 0, \quad \tilde{\lambda}\le 1$$

$$\tilde{\lambda}\frac{\partial F_2}{\partial\tilde{\lambda}} = C_3\left(e^{C_4\left(\tilde{\lambda}-1\right)}-1\right), \quad 1<\tilde{\lambda}<\lambda_m$$

$$\tilde{\lambda}\frac{\partial F_2}{\partial\tilde{\lambda}} = C_5+C_6\tilde{\lambda}, \quad \tilde{\lambda}\ge\lambda_m$$

Here, λ_m is the stretch at which the fibers are straightened, C_3 scales the exponential stresses, C_4 is the rate of uncrimping of the fibers, and C_5 is the modulus of the straightened fibers. C_6 is determined from the requirement that the stress is continuous at λ_m.

This material model uses a three-field element formulation, interpolating displacements as linear field variables and pressure and volume ratio as piecewise constant on each element [8].

In FEBio there is another transversely isotropic material model available that uses the Veronda-Westmann model as the material model for the isotropic ground substance matrix.

5.7. Biphasic Material

The biphasic material model differs significantly from the previous material models in that it also requires the explicit modeling of fluid that permeates the solid. The biphasic material model is useful to simulate materials that show viscoelastic behavior due to the presence of a fluid. Several biological materials such as cartilage can be described more accurately this way.

The finite element formulation for the biphasic material differs significantly from the standard solid displacement-only formulation in Chapter 3. This is a consequence of the need to introduce a new field variable, namely the fluid pressure *p*, which requires a coupled displacement-fluid pressure system to be solved. This section describes this formulation.

5.7.1. Governing Equations

The biphasic material is modeled as a mixture composed of a solid phase and a fluid phase. The equation of continuity for this mixture is given by

$$\nabla \cdot \left(\mathbf{v}^s + \mathbf{w} \right) = 0 , \tag{5.12}$$

where $\mathbf{v}^s$ is the solid fluid velocity and $\mathbf{w}$ is the *relative fluid flux*. The governing equations for the biphasic material are given by the balance of linear of momentum for both the solid and the fluid phase separately,

$$\nabla \cdot \boldsymbol{\sigma}^s + \boldsymbol{\pi} = 0 , \tag{5.13}$$

$$\nabla \cdot \boldsymbol{\sigma}^f - \boldsymbol{\pi} = 0 . \tag{5.14}$$

Here, $\boldsymbol{\pi}$ is the *drag force* and is the action-and-reaction interaction force between the solid and fluid phase. The specific form of the drag force is determined by a constitutive equation. In FEBio it is assumed to be given by

$$\boldsymbol{\pi} = \phi \mathbf{k}^{-1} \mathbf{w} , \tag{5.15}$$

where $\mathbf{k}$ is the *spatial permeability tensor*.

We further assume that the fluid is an inviscid Newtonian fluid. The final governing equations for a biphasic material are given by

$$\nabla \cdot \boldsymbol{\sigma} = \mathbf{0} , \tag{5.16}$$

$$\phi \nabla p + \boldsymbol{\pi} = \mathbf{0} . \tag{5.17}$$

Here $\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\sigma}^s$ is the apparent stress of the mixture. If we now substitute equation (5.15) for the drag force into equation (5.17) we get a relation between the relative fluid flux and the fluid pressure.

$$\mathbf{w} = -\mathbf{k} \cdot \nabla p \tag{5.18}$$

This relationship is also known as *Darcy's Law*.

In the presence of body forces, these equations modify to,

$$\boxed{\nabla \cdot \boldsymbol{\sigma} + \mathbf{f} = 0} \tag{5.19}$$

$$\boxed{\phi \left(\nabla p - \mathbf{f} \right) + \boldsymbol{\pi} = 0} \tag{5.20}$$

$$\boxed{\mathbf{w} = -\mathbf{k} \cdot \left(\nabla p - \mathbf{f} \right)} \tag{5.21}$$

Note that the spatial permeability tensor **k** in general is a function of the deformation. The exact dependence needs to be determined by a constitutive equation. It is easier however to state this constitutive equation in the material reference frame, and we therefore need to relate **k** to the *material permeability tensor* **K**. It can be shown that this relation is given by

$$\mathbf{K} = \frac{1}{J}\mathbf{F}^{-1}\mathbf{k}\mathbf{F}^{-T} . \tag{5.22}$$

5.7.2. Weak Formulation

Equations (5.19) and (5.20) specify the strong form of the equations of motion and continuity for a biphasic continuum. The finite element method is typically formulated using a weak, or variational formulation. The weak formulation is given by the virtual work equation,

$$\boxed{\delta W_m(\phi, p, \delta\mathbf{v}) = \int_v \boldsymbol{\sigma} : \delta\mathbf{d}dv - \int_v \mathbf{f}\cdot\delta\mathbf{v}dv - \int_{\partial v}\mathbf{t}\cdot\delta\mathbf{v}da = 0}, \tag{5.23}$$

and

$$\boxed{\delta W_f(\dot{\phi}, p, \delta p) = \int_v \left[\mathbf{w}\cdot\nabla(\delta p) - \delta p\mathbf{I}:\mathbf{d}\right]dv - \int_{\partial v}\delta p\mathbf{w}\cdot\mathbf{n}da = 0}. \tag{5.24}$$

The dependence of equation (5.24) on the pressure is implicit through the fluid flux term. For instance, for Darcy's Law we have $\mathbf{w} = -\mathbf{k}\cdot\nabla p$.

5.7.3. Finite Element Equations

In an iterative finite element procedure the equations (5.23) and (5.24) are linearized around the current configuration and discretized. The end result is a system of equations, given by

$$\begin{aligned} \mathbf{K}\Delta\mathbf{u} - \mathbf{G}\Delta p &= -\mathbf{R}_u \\ -\mathbf{G}^T\Delta\dot{\mathbf{u}} - \mathbf{Q}\Delta p &= -\mathbf{R}_p \end{aligned}, \tag{5.25}$$

where $\Delta\mathbf{u}$ is the displacement increment, $\Delta\dot{\mathbf{u}}$ the velocity increment and $\Delta\mathbf{p}$ is the pressure increment. To remove the dependence of the velocity increment, we apply the following approximation,

$$\Delta\dot{\mathbf{u}} \cong \frac{\Delta\mathbf{u}}{\Delta t}, \tag{5.26}$$

and obtain the final system of equations,

$$\boxed{\begin{bmatrix} \mathbf{K} & -\mathbf{G} \\ -\mathbf{G}^T & -\Delta t\mathbf{Q} \end{bmatrix}\begin{bmatrix} \Delta\mathbf{u} \\ \Delta p \end{bmatrix} = -\begin{bmatrix} \mathbf{R}_u \\ \Delta t\mathbf{R}_p \end{bmatrix}}. \tag{5.27}$$

K is the usual solid stiffness matrix, and **Q** and **G** are given by,

$$\mathbf{Q}^{(e)} = \int_{v^{(e)}} \mathbf{B}_p\mathbf{k}\mathbf{B}_p^T dv^{(e)}, \tag{5.28}$$

$$\mathbf{G}^{(e)} = \int_{v^{(e)}} \mathbf{B}_p \mathbf{N}^T dv\,, \tag{5.29}$$

where

$$\mathbf{B}_p = \begin{bmatrix} \nabla N_1 \\ \vdots \\ \nabla N_n \end{bmatrix} \quad \mathbf{N} = \begin{bmatrix} N_1 \\ \vdots \\ N_2 \end{bmatrix}. \tag{5.30}$$

Note that the total stiffness matrix in (5.27) is symmetric.

A major strength of this implementation of poroelasticity is the seamless integration of solid elements (with degrees of freedom only for the solid displacement) and biphasic elements (which also include degrees of freedom for fluid pressure), thanks to a judicious choice of primary variables and natural boundary conditions in the formulation of the governing equations. Thus, at the interface of a solid element and a biphasic element, continuity of the solid displacement is automatically satisfied, while the fluid flux across this interface automatically reduces to zero, since the solid element is impermeable by definition. In addition, any of the solid materials described in previous sections may be used to describe the solid phase.

5.8. Active Contraction Model

A time varying "elastance" active contraction model [12] was added to the transversely isotropic materials. When active contraction is activated, the total Cauchy stress $\boldsymbol{\sigma}$ is defined as the sum of the active stress tensor $\boldsymbol{\sigma}^a = T^a \boldsymbol{a} \otimes \boldsymbol{a}$ and the passive stress tensor $\boldsymbol{\sigma}^p$:

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^p + \boldsymbol{\sigma}^a, \tag{5.31}$$

where $\boldsymbol{a}$ is the deformed fiber vector (unit length), defined as $\lambda \boldsymbol{a} = \boldsymbol{F} \cdot \boldsymbol{a}_0$. The time varying elastance model is a modification of the standard Hill equation that scales the standard equation by an activation curve $C(t)$. The active fiber stress T^a is defined as:

$$T^a = T_{\max} \frac{Ca_0^2}{Ca_0^2 + ECa_{50}^2} C(t), \tag{5.32}$$

where T_{max} = 135.7 KPa is the isometric tension under maximal activation at the peak intracellular calcium concentration of Ca_0 = 4.35 μM. The length dependent calcium sensitivity is governed by the following equation:

$$ECa_{50} = \frac{(Ca_o)_{\max}}{\sqrt{\exp\left[B(l - l_0)\right] - 1}}, \tag{5.33}$$

where $(Ca_0)_{max}$ = 4.35 μM is the maximum peak intracellular calcium concentration, B = *4.75* μm^{-1} governs the shape of the peak isometric tension-sarcomere length relation, l_0 = 1.58 μm is the sarcomere length at which no active tension develops, and l is the sarcomere length which is the product of the fiber stretch λ and the sarcomere unloaded length l_r = *2.04* μm.

Chapter 6. Contact and Coupling

FEBio allows the user to connect the different parts of the model in various ways. Deformable parts can be connected to rigid bodies. Deformable objects can be brought in contact with each other. And finally rigid bodies can be connected with rigid joints. In this chapter we will describe these different ways to couple parts together.

6.1. Rigid-Deformable Coupling

In FEBio deformable meshes can be coupled with rigid bodies. The coupling requires a modification of the global stiffness matrix. Additionally, degrees of freedom need to be introduced for the rigid bodies [13]. This section describes the coupling between rigid and deformable bodies.

6.1.1. Kinematics

The position vector **x** of a finite element node may be denoted as,

$$\mathbf{x} = \mathbf{X} + \mathbf{u}\,, \tag{6.1}$$

where **X** is the initial position of the node and **u** the displacement vector. If this node is connected to a rigid body the position can alternatively be written as,

$$\mathbf{x} = \mathbf{r} + \mathbf{a}\,, \tag{6.2}$$

where **r** is the current position of the center of mass of the rigid body and **a** is the relative position of the node to the center of mass. The vector **a** may be written in terms of its initial value $\mathbf{a}_0$ in the undeformed state and a rotation matrix,

$$\mathbf{a} = \mathbf{Q}\mathbf{a}_0\,. \tag{6.3}$$

In an incremental displacement formulation equation (6.2) must be linearized:

$$\Delta\mathbf{u} = \Delta\mathbf{r} + \Delta\mathbf{Q}\mathbf{a}_0\,, \tag{6.4}$$

where the linearization of the rotation matrix can be expressed in a more convenient form,

$$\Delta\mathbf{Q} = \hat{\mathbf{a}}\Delta\boldsymbol{\theta}\,. \tag{6.5}$$

Here is $\Delta\boldsymbol{\theta}^T = \left[\Delta\theta_1, \Delta\theta_2, \Delta\theta_3\right]$ and the matrix $\hat{\mathbf{a}}$ is

$$\hat{\mathbf{a}} = \begin{bmatrix} 0 & a_3 & -a_2 \\ -a_3 & 0 & a_1 \\ a_2 & -a_1 & 0 \end{bmatrix}. \tag{6.6}$$

For a model containing both deformable and rigid nodes the nodal degrees of freedom may be grouped, and the above expressions used to obtain a condensed set of unknowns:

$$\begin{Bmatrix} \Delta\mathbf{u}^D \\ \Delta\mathbf{u}^R \end{Bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \hat{\mathbf{a}} \end{bmatrix} \begin{Bmatrix} \Delta\mathbf{u}^D \\ \Delta\mathbf{r} \\ \Delta\boldsymbol{\theta} \end{Bmatrix} = \mathbf{A}\Delta\tilde{\mathbf{u}}\,. \tag{6.7}$$

Substituting this into the discrete form of the principle of virtual work yields expressions for the condensed finite element stiffness matrix and residual vector for the coupled rigid-deformable system:

$$\tilde{\mathbf{K}}\Delta\tilde{\mathbf{u}} = -\tilde{\mathbf{R}}, \quad \tilde{\mathbf{K}} = \mathbf{A}^T\mathbf{K}\mathbf{A}, \quad \tilde{\mathbf{R}} = \mathbf{A}^T\mathbf{R}\,. \tag{6.8}$$

6.1.2. A single rigid body

The global system of equations can now be written as follows (for a single rigid body coupled to a deformable body),

$$\begin{bmatrix} \mathbf{K}^D & \mathbf{K}^{DR} \\ \left(\mathbf{K}^{DR}\right)^T & \mathbf{K}^R \end{bmatrix} \begin{Bmatrix} \Delta\mathbf{u}^D \\ \Delta\mathbf{r} \\ \Delta\boldsymbol{\theta} \end{Bmatrix} = -\begin{Bmatrix} \mathbf{R}^D \\ \mathbf{F}^R \\ \mathbf{M}^R \end{Bmatrix}. \tag{6.9}$$

Here $\mathbf{F}^R$ is formed by adding all the residual vectors of all interface nodes that connect the deformable body to the rigid body,

$$\mathbf{F}^R = \sum_i \mathbf{R}_i^D , \tag{6.10}$$

where *i* sums over all interface nodes, and

$$\mathbf{M}^R = \sum_i \hat{\mathbf{a}}_i \mathbf{R}_i^D . \tag{6.11}$$

It is recognized that $\mathbf{F}^R$ is simply the total residual force that is applied to the rigid body and $\mathbf{M}^R$ is the total residual torque.

Constructing the stiffness matrix is accomplished in a similar manner. Assume *n* nodes per element, than the normal element stiffness matrix (in absence of rigid nodes) is given by,

$$\mathbf{k}^{(e)} = \begin{bmatrix} \mathbf{k}_{11} & \cdots & \mathbf{k}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{k}_{n1} & \cdots & \mathbf{k}_{nn} \end{bmatrix}, \tag{6.12}$$

where $\mathbf{k}_{ij}$ is the nodal stiffness matrix connecting node *i* to node *j*. These nodal stiffness matrices are now assembled into the global stiffness matrix. If node *i* and *j* are neither interface nodes their nodal stiffness matrix is assembled into $\mathbf{K}^D$ in the usual manner,

$$\mathbf{K}^D = \sum_e \mathbf{k}_{ij}^{(e)} , \tag{6.13}$$

where the sum now has to interpreted as the finite element assembly operator.

If node *j* is an interface node, than the nodal stiffness matrix gets assembled in the $\mathbf{K}^{DR}$ matrix:

$$\mathbf{K}^{DR} = \sum_e \begin{bmatrix} \mathbf{k}_{ij}^{(e)} & \mathbf{k}_{ij}^{(e)}\hat{\mathbf{a}}_j \end{bmatrix}. \tag{6.14}$$

If both nodes belong to the rigid body than the nodal element matrix gets assembled in $\mathbf{K}^R$ as follows,

$$\mathbf{K}^R = \sum_e \begin{bmatrix} \mathbf{k}_{ij}^{(e)} & \mathbf{k}_{ij}^{(e)}\hat{\mathbf{a}}_j \\ \hat{\mathbf{a}}_i^T\mathbf{k}_{ij}^{(e)T} & \hat{\mathbf{a}}_i^T\mathbf{k}_{ij}^{(e)}\hat{\mathbf{a}}_j \end{bmatrix}. \tag{6.15}$$

6.1.3. Multiple Rigid Bodies

The previous results can easily be extended if there are multiple rigid bodies. The following section presents the approach for two rigid bodies, but the results can easily be generalized to *N* rigid bodies.

For two rigid bodies, the global system of equations takes the following form,

$$\begin{bmatrix} \mathbf{K}^D & \mathbf{K}_1^{DR} & \mathbf{K}_2^{DR} \\ \left(\mathbf{K}_1^{DR}\right)^T & \mathbf{K}_{11}^R & \mathbf{K}_{12}^R \\ \left(\mathbf{K}_2^{DR}\right)^T & \mathbf{K}_{21}^R & \mathbf{K}_{22}^R \end{bmatrix} \begin{Bmatrix} \Delta\mathbf{u}^D \\ \Delta\mathbf{r}_1 \\ \Delta\boldsymbol{\theta}_1 \\ \Delta\mathbf{r}_2 \\ \Delta\boldsymbol{\theta}_2 \end{Bmatrix} = -\begin{Bmatrix} \mathbf{R}^R \\ \mathbf{F}_1 \\ \mathbf{M}_1 \\ \mathbf{F}_2 \\ \mathbf{M}_2 \end{Bmatrix}. \tag{6.16}$$

Care must be taken to assemble the nodal stiffness matrix in the correct global sub-matrix. If node *i* is not an interface node and node *j* is connected to rigid body 1, then their nodal stiffness matrix goes into $\mathbf{K}_1^{DR}$. If, however, node *j* is attached to rigid body 2 then their nodal stiffness matrix goes into $\mathbf{K}_2^{DR}$. If node *i* is connected to rigid body 1 and node *j* is connected to rigid body 2, then their nodal stiffness matrix goes into $\mathbf{K}_{12}^R$, and so on. Note that it is assumed here that a node may only connect to a single rigid body. Coupling rigid bodies through joints is described in the next section.

6.2. Sliding Interfaces

This section summarizes the theoretical developments of the two body contact problem. After introducing some notation and terminology, the contact integral is presented, which contains the contribution to the virtual work equation from the contact tractions. Since the nonlinear contact problem is solved using a Newton based iterative method, the contact integral is linearized. Next, anticipating a finite element implementation, the contact integral and its linearization are discretized using a standard finite element approach. Finally the augmented Lagrangian method for enforcing the contact constraints is described.

6.2.1. Contact Kinematics

For the most part the notation of this section follows [14], with a few simplifications here and there since the implementation in FEBio is currently for quasi-static, frictionless, two body contact problem.

The volume occupied by body *i* in the reference configuration is denoted by $\Omega^{(i)} \subset \mathbb{R}^3$ where $i = 1,2$. The boundary of body *i* is denoted by $\Gamma^{(i)}$ and is divided into three regions $\Gamma^{(i)} = \Gamma_\sigma^{(i)} \bigcup \Gamma_u^{(i)} \bigcup \Gamma_c^{(i)}$, where $\Gamma_\sigma^{(i)}$ is the boundary where tractions are applied, $\Gamma_u^{(i)}$ the boundary where the solution is prescribed and $\Gamma_c^{(i)}$ the part of the

boundary that will be in contact with the other body. It is assumed that $\Gamma_\sigma^{(i)} \bigcap \Gamma_u^{(i)} \bigcap \Gamma_c^{(i)} = \varnothing$.

The deformation of body *i* is defined by $\varphi^{(i)}$. The boundary of the deformed body *i*, that is the boundary of $\varphi^{(i)}\left(\Omega^{(i)}\right)$ is denoted by $\gamma^{(i)} = \gamma_\sigma^{(i)} \bigcup \gamma_u^{(i)} \bigcup \gamma_c^{(i)}$ where $\gamma_\sigma^{(i)} = \varphi^{(i)}\left(\Gamma_\sigma^{(i)}\right)$ is the boundary in the current configuration where the tractions are applied and similar definitions for $\gamma_u^{(i)}$ and $\gamma_c^{(i)}$. See the figure below for a graphical illustration of the defined regions.

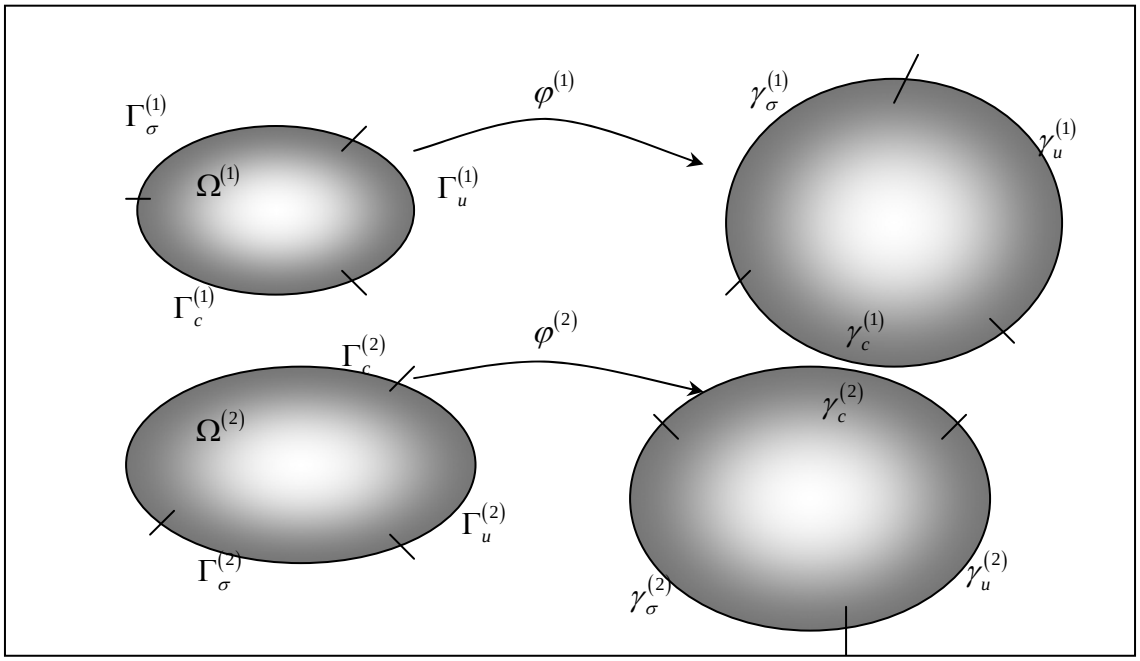

Figure 6-1. The two-body contact problem.

Points in body 1 are denoted by **X** in the reference configuration and **x** in the current configuration. For body 2 these points are denoted by **Y** and **y**. To define contact, the location where the two bodies are in contact with each other must be established. If body 1 is the *slave body* and body 2 is the *master body*, then for a given point **X** on the slave reference contact surface there is a point $\bar{\mathbf{Y}}(\mathbf{X})$ on the master contact surface that is in some sense closest to point **X**. This closest point is defined in a closest point projection sense:

$$\boxed{\bar{\mathbf{Y}}(\mathbf{X}) = \arg\min_{\mathbf{Y}\in\Gamma_c^{(2)}} \left\| \varphi^{(1)}(\mathbf{X}) - \varphi^{(2)}(\mathbf{Y}) \right\|}. \tag{6.17}$$

With the definition of $\bar{\mathbf{Y}}(\mathbf{X})$ established the *gap function* can be defined, which is a measure for the distance between **X** and $\bar{\mathbf{Y}}(\mathbf{X})$,

$$\boxed{g(\mathbf{X}) = -\mathbf{v}\cdot\left(\varphi^{(1)}(\mathbf{X}) - \varphi^{(2)}\left(\bar{\mathbf{Y}}(\mathbf{X})\right)\right)}, \tag{6.18}$$

where $\mathbf{v}$ is the local surface normal of surface $\gamma_c^{(2)}$ evaluated at $\bar{\mathbf{y}} = \varphi^{(2)}\left(\bar{\mathbf{Y}}\left(\mathbf{X}\right)\right)$. Note that $g > 0$ when $\mathbf{X}$ has penetrated body 2, so that the constraint condition to be satisfied at all time is $g \le 0$.

6.2.2. Weak Form of Two Body Contact

The balance of linear momentum can be written for each of the two bodies in the reference configuration,

$$G^{(i)}\left(\varphi^{(i)}, w^{(i)}\right) = \int_{\Omega^{(i)}} GRAD\left[w^{(i)}\right] : \mathbf{P}^{(i)} d\Omega - \int_{\Omega^{(i)}} w^{(i)} \cdot \mathbf{F}^{(i)} d\Omega - \int_{\Gamma_s^{(i)}} w^{(i)} \cdot \mathbf{T}^{(i)} d\Gamma - \int_{\Gamma_c^{(i)}} w^{(i)} \cdot \mathbf{T}^{(i)} d\Gamma = 0 \tag{6.19}$$

where $w^{(i)}$ is a weighting function and $\mathbf{P}$ is the 1st Piola-Kirchhoff stress tensor. The last term corresponds to the virtual work of the contact tractions on body *i*. For notational convenience, the notations φ and w are introduced to denote the collection of the respective mappings $\varphi^{(i)}$ and $w^{(i)}$ (for *i*=1,2). In other words,

$$\begin{aligned} \varphi &: \bar{\Omega}^{(1)} \bigcup \bar{\Omega}^{(2)} \to \mathbb{R}^3 \\ w &: \bar{\Omega}^{(1)} \bigcup \bar{\Omega}^{(2)} \to \mathbb{R}^3 \end{aligned}. \tag{6.20}$$

The variational principle for the two body system is the sum of (6.19) for body 1 and 2 and can be expressed as,

$$\begin{aligned} G\left(\varphi, w\right) &:= \sum_{i=1}^{2} G^{(i)}\left(\varphi^{(i)}, w^{(i)}\right) \\ &= \underbrace{\sum_{i=1}^{2} \left\{ \int_{\Omega^{(i)}} GRAD\left[w^{(i)}\right] : \mathbf{P}^{(i)} d\Omega - \int_{\Omega^{(i)}} w^{(i)} \cdot \mathbf{F}^{(i)} d\Omega - \int_{\Gamma_s^{(i)}} w^{(i)} \cdot \mathbf{T}^{(i)} d\Gamma \right\}}_{G^{\text{int,ext}}\left(\varphi, w\right)} \\ &\quad \underbrace{-\sum_{i=1}^{2} \int_{\Gamma_c^{(i)}} w^{(i)} \cdot \mathbf{T}^{(i)} d\Gamma}_{G^{c}\left(\varphi, w\right)} \end{aligned} \tag{6.21}$$

Or in short,

$$\boxed{G\left(\varphi, w\right) = G^{\text{int,ext}}\left(\varphi, w\right) + G^{c}\left(\varphi, w\right)}. \tag{6.22}$$

Note that the minus sign is included in the definition of the contact integral G^c. The contact integral can be written as an integration over the contact surface of body 1 by balancing linear momentum across the contact surface:

$$\mathbf{t}^{(2)}\left(\bar{\mathbf{y}}\left(\mathbf{x}\right)\right) d\Gamma^{(2)} = -\mathbf{t}^{(1)}\left(\mathbf{x}\right) d\Gamma^{(1)}. \tag{6.23}$$

The contact integral can now be rewritten over the contact surface of body 1:

$$G^{c} = -\int_{\Gamma_c^{(1)}} \mathbf{t}^{(1)}\left(\mathbf{x}\right) \cdot \left[w^{(1)}\left(\mathbf{x}\right) - w^{(2)}\left(\bar{\mathbf{y}}\left(\mathbf{x}\right)\right) \right] d\Gamma. \tag{6.24}$$

In the case of frictionless contact, the contact traction is taken as perpendicular to surface 2 and therefore can be written as, $\mathbf{t}^{(1)} = t_N \mathbf{v}$ where $\mathbf{v}$ is the (outward) surface normal

and t_N is to be determined from the solution strategy. For example in a Lagrange multiplier method the t_N 's would be the Lagrange multipliers.

By noting that the variation of the gap function is given by

$$\delta g = -\mathbf{v}\cdot\left(w^{(1)}(\mathbf{x}) - w^{(2)}\left(\bar{\mathbf{y}}(\mathbf{x})\right)\right), \tag{6.25}$$

equation (6.24) can be simplified as,

$$\boxed{G^c = \int_{\Gamma_c^{(1)}} t_N \delta g d\Gamma}. \tag{6.26}$$

6.2.3. Linearization of the Contact Integral

In a Newton-Raphson implementation the contact integral must be linearized with respect to the current configuration:

$$\Delta G^c(\varphi, w) = \int_{\Gamma_c^{(1)}} \Delta\left(t_N \delta g\right) d\Gamma\,. \tag{6.27}$$

Examining the normal contact term first, the directional derivative of t_N is given (for the case of the penalty regularization) by:

$$\begin{aligned} \Delta t_N &= \Delta\left\{\varepsilon_N \langle g \rangle\right\} \\ &= H(g)\varepsilon_N \Delta g \end{aligned}, \tag{6.28}$$

where ε_N is the penalty factor and $H(g)$ is the Heaviside function. The quantity $\Delta(\delta g)$ is given by,

$$\begin{aligned} \Delta(\delta g) = {} & g\left[\mathbf{v}\cdot\delta\varphi^{(2)}_{,\gamma}\left(\bar{\mathbf{Y}}(\mathbf{X})\right) + \kappa_{\alpha\gamma}\left(\bar{\mathbf{Y}}(\mathbf{X})\right)\delta\bar{\xi}_\alpha\right] m^{\gamma\beta} \\ & \left[\mathbf{v}\cdot\Delta\varphi^{(2)}_{,\beta}\left(\bar{\mathbf{Y}}(\mathbf{X})\right) + \kappa_{\alpha\beta}\left(\bar{\mathbf{Y}}(\mathbf{X})\right)\Delta\bar{\xi}^\alpha\right] \\ & + \delta\bar{\xi}^\beta \mathbf{v}\cdot\left[\Delta\varphi^{(2)}_{,\beta}\left(\bar{\mathbf{Y}}(\mathbf{X})\right)\right] + \Delta\bar{\xi}^\beta \mathbf{v}\cdot\left[\delta\varphi^{(2)}_{,\beta}\left(\bar{\mathbf{Y}}(\mathbf{X})\right)\right] \\ & + \kappa_{\alpha\beta}\left(\bar{\mathbf{Y}}(\mathbf{X})\right)\delta\bar{\xi}^\beta \Delta\bar{\xi}^\alpha \end{aligned}. \tag{6.29}$$

6.2.4. Discretization of the Contact Integral

The contact integral, which is repeated here,

$$G^c(\varphi, w) = \int_{\Gamma^{(1)}} t_N \delta g d\Gamma\,, \tag{6.30}$$

will now be discretized using a standard finite element procedure. First it is noted that the integration can be written as a sum over the surface element areas:

$$G^c(\varphi, w) = \sum_{e=1}^{N_{sel}} \int_{\Gamma^{(1)e}} t_N \delta g d\Gamma\,, \tag{6.31}$$

where N_{sel} is the number of surface elements. The integration can be approximated using a quadrature rule,

$$G^c(\varphi,w) \cong \sum_{e=1}^{N_{sel}} \left\{ \sum_{i=1}^{N_{\text{int}}^e} w^i j(\xi_i) t_N(\xi_i) \delta g(\xi_i) \right\}, \tag{6.32}$$

where N_{int}^e are the number of integration points for element *e*. It is now assumed that the integration points coincide with the element's nodes (e.g. for a quadrilateral surface element we have $\xi_1 = (-1,-1)$, $\xi_2 = (1,-1)$, $\xi_3 = (1,1)$ and $\xi_4 = (-1,1)$). With this quadrature rule, we have

$$\begin{aligned} w^{(1)}(\xi_i) &= \mathbf{c}_i^{(1)} \\ w^{(2)}(\bar{\xi}_i) &= \sum_{j=1}^{n} N_j(\bar{\xi}_i)\mathbf{c}_j^{(2)} \end{aligned}, \tag{6.33}$$

so that,

$$\delta g(\xi_i) = -\mathbf{v} \cdot \left(\mathbf{c}_i^{(1)} - \sum_{j=1}^{n} N_j^{(2)}(\bar{\xi}_i)\mathbf{c}_j^{(2)} \right). \tag{6.34}$$

If the following vectors are defined,

$$\begin{aligned} \delta\mathbf{\Phi}^T &= \left[\mathbf{c}_i^{(1)}, \quad \mathbf{c}_1^{(2)}, \quad \cdots, \quad \mathbf{c}_n^{(2)} \right] \\ \mathbf{N}^T &= \left[\mathbf{v}, \quad -\mathbf{v}N_1^{(2)}, \quad \cdots, \quad -\mathbf{v}N_n^{(2)} \right] \end{aligned}, \tag{6.35}$$

equation (6.32) can then be rewritten as follows,

$$\boxed{G^c(\varphi,\delta\mathbf{v}) \cong \sum_{e=1}^{N_{sel}} \left\{ \sum_{i=1}^{N_{\text{int}}^e} w^i j(\xi_i) t_N(\xi_i) \delta\mathbf{\Phi}^T \mathbf{N}^T \right\}}. \tag{6.36}$$

The specific form for t_N will depend on the method employed for enforcing the contact constraint.

6.2.5. Discretization of the Contact Stiffness

A similar procedure can now be used to calculate the discretized contact stiffness matrix. The linearization of the contact integral is repeated here:

$$\begin{aligned} \Delta G^c(\varphi,w) &= \sum_{e=1}^{N_{sel}} \int_{\Gamma^{(1)e}} \Delta(t_N \delta g)\, d\Gamma \\ &= \sum_{e=1}^{N_{sel}} \sum_{i=1}^{N_{\text{int}}^e} w_i j(\xi_i) \Delta(t_N \delta g)(\xi_i) \end{aligned}. \tag{6.37}$$

Using matrix notation we can rewrite equation (6.37) as,

$$\Delta W^c(\varphi,\delta\mathbf{v}) = \sum_{e=1}^{N_{sel}} \sum_{i}^{N_{\text{int}}^e} w_i j(\xi_i) \delta\mathbf{\Phi} \cdot \mathbf{k}^c \Delta\mathbf{\Phi}, \tag{6.38}$$

where $\delta\mathbf{\Phi}$ is as above and $\Delta\mathbf{\Phi}$ similar to $\delta\mathbf{\Phi}$ with δ replaced with Δ and $\mathbf{k}^c$,

$$\begin{aligned}\mathbf{k}^c = \varepsilon_N H\left(\lambda_N^k + \varepsilon_N g\right)\mathbf{N}\mathbf{N}^T + t_N\Big\{g\Big[m^{11}\bar{\mathbf{N}}_1\bar{\mathbf{N}}_1^T \\ +m^{12}\left(\bar{\mathbf{N}}_1\bar{\mathbf{N}}_2^T + \bar{\mathbf{N}}_2\bar{\mathbf{N}}_1^T\right) + m^{22}\bar{\mathbf{N}}_2\bar{\mathbf{N}}_2^T\Big] - \mathbf{D}_1\mathbf{N}_1^T \\ -\mathbf{D}_2\mathbf{N}_2^T - \mathbf{N}_1\mathbf{D}_1^T - \mathbf{N}_2\mathbf{D}_2^T + \kappa_{12}\left(\mathbf{D}_1\mathbf{D}_2^T + \mathbf{D}_2\mathbf{D}_1^T\right)\Big\}\end{aligned}, \tag{6.39}$$

where,

$$\mathbf{N} = \begin{bmatrix}\mathbf{v}\\ -N_1\left(\bar{\boldsymbol{\xi}}\right)\mathbf{v}\\ \vdots\\ -N_4\left(\bar{\boldsymbol{\xi}}\right)\mathbf{v}\end{bmatrix},\quad \mathbf{T}_\alpha = \begin{bmatrix}\boldsymbol{\tau}_\alpha\\ -N_1\left(\bar{\boldsymbol{\xi}}\right)\boldsymbol{\tau}_\alpha\\ \vdots\\ -N_4\left(\bar{\boldsymbol{\xi}}\right)\boldsymbol{\tau}_\alpha\end{bmatrix},\quad \mathbf{N}_\alpha = \begin{bmatrix}\mathbf{0}\\ -N_{1,\alpha}\left(\bar{\boldsymbol{\xi}}\right)\mathbf{v}\\ \vdots\\ -N_{4,\alpha}\left(\bar{\boldsymbol{\xi}}\right)\mathbf{v}\end{bmatrix}. \tag{6.40}$$

The following vectors are also defined which depend on the vectors of (6.40):

$$\begin{aligned}\mathbf{D}_1 &= \frac{1}{\det\mathbf{A}}\left[A_{22}\left(\mathbf{T}_1 + g\mathbf{N}_1\right) - A_{12}\left(\mathbf{T}_2 + g\mathbf{N}_2\right)\right]\\ \mathbf{D}_2 &= \frac{1}{\det\mathbf{A}}\left[A_{11}\left(\mathbf{T}_2 + g\mathbf{N}_2\right) - A_{12}\left(\mathbf{T}_1 + g\mathbf{N}_1\right)\right],\\ \bar{\mathbf{N}}_1 &= \mathbf{N}_1 - \kappa_{12}\mathbf{D}_2\\ \bar{\mathbf{N}}_2 &= \mathbf{N}_2 - \kappa_{12}\mathbf{D}_1\end{aligned} \tag{6.41}$$

where the matrix **A** is defined as,

$$A_{ij} = m_{ij} + g\kappa_{ij}\,. \tag{6.42}$$

Here, $m_{ij} = \boldsymbol{\tau}_i \cdot \boldsymbol{\tau}_j$ is the surface metric tensor and $\kappa_{ij} = \mathbf{v}\cdot\varphi_t^{(2)}{}_{,ij}\left(\bar{\mathbf{Y}}\right)$ denotes the components of the surface curvature at $\bar{\boldsymbol{\xi}}$.

6.2.6. Augmented Lagrangian Method

The augmented Lagrangian method is used in FEBio to enforce the contact constraints to a user-specified tolerance. This implies that the normal contact tractions are given by,

$$t_N = \left\langle \lambda_N + \varepsilon_N g\right\rangle. \tag{6.43}$$

Note that this assumption is consistent with the approach that was used in establishing the discretization of the linearization of the contact integral (6.39). In (6.43) ε_N is a penalty factor that is chosen arbitrarily.

The Newton-Raphson iterative method is now used to solve the nonlinear contact problem where Uzawa's method (REF) is employed to calculate the Lagrange multipliers λ_N. This implies that the Lagrange multipliers are kept fixed during the Newton-Raphson iterations. After convergence the multipliers are updated and a new NR procedure is started. This procedure can be summarized by the following four steps.

1. **Initialize** the augmented Lagrangian iteration counter *k*, and the initial guesses for the multipliers:

$$\begin{aligned} &\lambda_{N_{n+1}}^{(0)} = \lambda_{N_n} \\ &k = 0 \end{aligned} . \tag{6.44}$$

2. **Solve** for $\mathbf{d}_{n+1}^{(k)}$, the solution vector corresponding to the fixed *k*th iterate for the multipliers,

$$\mathbf{F}^{\text{int}}\left(\mathbf{d}_{n+1}^{(k)}\right) + \mathbf{F}^{c}\left(\mathbf{d}_{n+1}^{(k)}\right) = \mathbf{F}_{n+1}^{ext}, \tag{6.45}$$

where the contact tractions used to compute $\mathbf{F}^{c}$, the contact force, are governed by

$$t_{N_{n+1}}^{(k)} = \left\langle \lambda_{N_{n+1}}^{(k)} + \varepsilon_N g_{n+1}^{k} \right\rangle . \tag{6.46}$$

3. **Update** the Lagrange multipliers and iteration counters:

$$\begin{aligned} &\lambda_{N_{n+1}}^{(k+1)} = \left\langle \lambda_{N_{n+1}}^{(k)} + \varepsilon_N g_{n+1}^{(k)} \right\rangle \\ &k = k+1 \end{aligned} . \tag{6.47}$$

4. **Return** to the solution phase.

Steps 2-4 of the above algorithm are generally repeated until all contact constraints are satisfied to a user-specified tolerance or little change in the solution vector from augmentation to augmentation is noted.

References

[1] Spencer, A. J. M., 1984, Continuum Theory of the Mechanics of Fibre-Reinforced Composites, Springer-Verlag, New York.
[2] Bonet, J., and Wood, R. D., 1997, Nonlinear continuum mechanics for finite element analysis, Cambridge University Press.
[3] Weiss, J. A., Maker, B. N., and Govindjee, S., 1996, "Finite element implementation of incompressible, transversely isotropic hyperelasticity," Computer Methods in Applications of Mechanics and Engineering, 135, pp. 107-128.
[4] Horowitz, A., Sheinman, I., Lanir, Y., Perl, M., and Sideman, S., 1988, "Nonlinear Incompressilbe Finite Element for Simulating Loading of Cardiac Tissue-part I: two Dimensional Formulation for Thin Myocardial Strips," Journal of Biomechanical Engineering, Transactions of the ASME, 110(1), pp. 57-61.
[5] Humphrey, J. D., Strumpf, R. K., and Yin, F. C. P., 1990, "Determination of a constitutive relation for passive myocardium. I. A new functional form," Journal of Biomechanical Engineering, Transactions of the ASME, 112(3), pp. 333-339.
[6] Humphrey, J. D., and Yin, F. C. P., 1987, "On constitutive Relations and Finite Deformations of Passive Cardiac Tissue: I. A Pseudostrain-Energy Function," Journal of Biomechanical Engineering, Transactions of the ASME, 109(4), pp. 298-304.
[7] Matthies, H., and Strang, G., 1979, "The solution of nonlinear finite element equations," Intl J Num Meth Eng, 14, pp. 1613-1626.
[8] Simo, J. C., and Taylor, R. L., 1991, "Quasi-incompressible finite elasticity in principal stretches: Continuum basis and numerical algorithms," Computer Methods in Applied Mechanics and Engineering, 85, pp. 273-310.
[9] Veronda, D. R., and Westmann, R. A., 1970, "Mechanical Characterization of Skin - Finite Deformations," J. Biomechanics, Vol. 3, pp. 111-124.
[10] Puso, M. A., and Weiss, J. A., 1998, "Finite element implementation of anisotropic quasi-linear viscoelasticity using a discrete spectrum approximation," J Biomech Eng, 120(1), pp. 62-70.
[11] Quapp, K. M., and Weiss, J. A., 1998, "Material characterization of human medial collateral ligament," J Biomech Eng, 120(6), pp. 757-763.
[12] Guccione, J. M., and McCulloch, A. D., 1993, "Mechanics of active contraction in cardiac muscle: part I - constitutive relations for fiber stress that describe deactivation," J. Biomechanical Engineering, vol. 115(no. 1), pp. 72-83.
[13] Maker, B. N., 1995, "Rigid bodies for metal forming analysis with NIKE3D," University of California, Lawrence Livermore Lab Rept, UCRL-JC-119862, pp. 1-8.
[14] Laursen, T. A., 2002, Computational Contact and Impact Mechanics, Springer.