DEVELOPMENT OF A NONLINEAR FINITE ELEMENT CODE FOR COMPUTATIONAL BIOMECHANICS AND BIOPHYSICS

by

Steve Maas

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Computing

School of Computing

The University of Utah

May 2017

The University of Utah Graduate School

STATEMENT OF DISSERTATION APPROVAL

The dissertation of **Steve Maas**

has been approved by the following supervisory committee members:

Jeffrey A. Weiss	, Chair	**1/26/2017** Date Approved
Robert Michael Kirby II	, Member	**1/26/2017** Date Approved
Martin Berzins	, Member	**1/30/2017** Date Approved
James Guilkey	, Member	**1/26/2017** Date Approved
Gerard A. Ateshian	, Member	**1/20/2017** Date Approved

and by **Ross Whitaker** , Chair of

the School of **Computing**

and by David B. Kieda, Dean of The Graduate School.

ABSTRACT

Despite the progress that has been made since the inception of the finite element method, the field of biomechanics has generally relied on software tools that were not specifically designed to target this particular area of application. Software designed specifically for the field of computational biomechanics does not appear to exist. To overcome this limitation, FEBio was developed, an acronym for “Finite Elements for Biomechanics”, which provided an open-source framework for developing finite element software that is tailored to the specific needs of the biomechanics and biophysics communities. The proposed work added an extendible framework to FEBio that greatly facilitates the implementation of novel features and provides an ideal platform for exploring novel computational approaches. This framework supports plugins, which simplify the process of adding new features even more since plugins can be developed independently from the main source code. Using this new framework, this work extended FEBio in two important areas of interest in biomechanics. First, as tetrahedral elements continue to be the preferred modeling primitive for representing complex geometries, several tetrahedral formulations were investigated in terms of their robustness and accuracy for solving problems in computational biomechanics. The focus was on the performance of quadratic tetrahedral formulations in large deformation contact analyses, as this is an important area of application in biomechanics. Second, the application of prestrain to computational models has been recognized as an important component in simulations of biological tissues in order to accurately predict the mechanical response. As this remains challenging to do in existing software packages, a general computational framework for applying prestrain was incorporated in the FEBio software. The work demonstrated via several examples how plugins greatly simplify the development of novel features. In addition, it showed that the quadratic tetrahedral formulations studied in this work are viable alternatives for contact analyses. Finally, it demonstrated the newly developed prestrain plugin and showed how it can be used in various applications of prestrain.

TABLE OF CONTENTS

ABSTRACT……………………………………………………………………...…iii

LIST OF FIGURES…………………………………………………………...………vii

ACKNOWLEDGEMENTS…………………………………………………………....ix

Chapters

1. INTRODUCTION 1

1.1 Motivation.. 1
1.2 Research Goals.. 12
1.3 Summary of Chapters 13
1.4 References.. 15

2. FEBIO: FINITE ELEMENTS FOR BIOMECHANICS 23

2.1 Abstract 23
2.2 Introduction.. 24
2.3 Overview.. 27
2.4 Theory and Implementation.. 29
2.5 Verification Problems 35
2.6 Discussion 50
2.7 Acknowledgments.. 52
2.8 References.. 52

3. A PLUGIN FRAMEWORK FOR EXTENDING THE SIMULATION CAPABILITIES OF FEBIO 57

3.1 Abstract 57
3.2 Introduction.. 58
3.3 Methods.. 61
3.4 Results.. 69
3.5 Discussion 71
3.6 References.. 73

4. FINITE ELEMENT SIMULATION OF ARTICULAR CONTACT MECHANICS WITH QUADRATIC TETRAHEDRAL ELEMENTS... 76

4.1 Abstract 76
4.2 Introduction 77
4.3 Methods 80
4.4 Results 87
4.5 Discussion 94
4.6 Acknowledgments 98
4.7 References 98

5. ADDITIONAL SIMULATIONS USING QUADRATIC TETRAHEDRAL ELEMENTS 102

5.1 Introduction 102
5.2 Methods 103
5.3 Results 107
5.4 Discussion 111

6. A GENERAL FRAMEWORK FOR APPLICATION OF PRESTRAIN TO COMPUTATIONAL MODELS OF BIOLOGICAL MATERIALS 114

6.1 Abstract 114
6.2. Introduction 115
6.3. Methods 118
6.4 Results 133
6.5. Discussion 142
6.6 Acknowledgments 145
6.7 Appendix 146
6.8 References 147

7. SUMMARY AND DISCUSSION 151

7.1 Summary 151
7.2 Future Work 159
7.3 References 166

8. APPLICATIONS OF FEBIO 170

8.1 Introduction 170
8.2 PostView and PreView 170
8.3 FEBio Material Library 171
8.4 Material Parameter Optimization 172
8.5 FEBio Contact 173
8.6 Computational Homogenization 173
8.7 Multiphasic Framework 174
8.8 References 175

LIST OF FIGURES

Figures Page

2.1: Simulation of uniaxial tension-compression material tests 37

2.2: Confined compression creep testing of deformable porous media...................... 38

2.3: Unconfined compression stress relaxation. 39

2.4: Strip biaxial stretching of an elastic sheet with a circular hole. 41

2.5: Geometrically nonlinear analysis of a cantilever beam...................................... 42

2.6: Twisted ribbon test for shells... 44

2.7: Upsetting of an elastic billet. 45

2.8: Crushing of a pipe.. 47

2.9: Cartilage layer compressed by a flat, rigid, impermeable surface....................... 49

2.10: Validated model of the human inferior glenohumeral ligament,....................... 50

3.1: Hierarchy of FEBio modules.. 61

3.2: Stress evaluation for the neo-Hookean material... 63

3.3: The Angio plugin couples the Angio code, a library that simulates the growth of blood vessels, with FEBio 68

3.4: Prestrain analysis of dogbone sample, using a transversely isotropic material... 69

3.5: 2D slice from a 3D warping analysis that tracks the deformation of the left ventricle.. 70

3.6: Simulation of angiogenesis.. 71

4.1: Schematic of the node topology for two quadratic tetrahedral elements............. 80

4.2: Schematic of the surface integration rules 82

4.3: Schematic of the plane strain contact problem. 84

4.4: Geometry and loading path of the rigid cylinder on the deformable box. 86

4.5: Results for the plane strain contact problem. 88

4.6: Average reaction force on the rigid indenter 89

4.7: Variation of the reaction force on the rigid indenter 91

4.8: 3rd principal stress fringe plots for two different mesh densities 92

4.9: Finite element model of the human hip 93

5.1: Comparison of contact stress predictions for the highest mesh resolution 107

5.2: Comparison of the tangential (X) component of the reaction force 108

5.3: Comparison of contact stresses between structured and unstructured meshes.. 109

5.4: Contact stresses for the unaligned indenter problem 110

5.5: Contact stresses near the area of indentation for various element formulations. 111

5.6: Reaction force as a function of the total number of nodes in the mesh 112

6.1: Schematic showing the prestressed reference configuration and the loaded, deformed configuration. 118

6.2: Flowchart of numerical algorithm 123

6.3: Verification example of the prestrain gradient approach 135

6.4: Effects of applying *in situ* stretch to a reference geometry that is kinematically incompatible. 137

6.5: Example illustrating the method to recover the displacement map from the deformation gradient 139

6.6: Finite element analysis of the Open Knee model 141

7.1. Comparison of forward and inverse FE problem. 166

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to several people who have had a profound influence on my education and research over the past years. First and foremost, I am indebted to my PhD advisor, Dr. Jeffrey Weiss. It’s been more than ten years since he introduced me to the field of computational biomechanics, but my interest in this domain continues to grow unabated. Our shared vision for a novel finite element tool in this field led us down the path to FEBio and provided the impetus for this dissertation. I do not exaggerate when I say that of all the projects I have worked on, I was and still am most excited about the FEBio project. And I am extremely grateful that Dr. Weiss provided the opportunity for me to take a leading role in this project. I would also like to thank Dr. Gerard Ateshian. Not only has he contributed to the FEBio project in a profound and significant way, his dedication and excitement and the many insightful discussions we’ve had over the years have been inspiring. Finally, I want to thank all the members of the Musculoskeletal Research Laboratory, in particular Dr. Benjamin Ellis and David Rawlins, as well as the many graduate and undergraduate students, who have helped create a joyful and stimulating work and research environment.

I would also like to acknowledge the financial support that allowed me to pursue this research. I am in particular grateful for the following grants provided by the National Institutes of Health: R01GM083925, R01GM104139, R25GM107009, R01EB016701, and R01EB015133.

CHAPTER 1

INTRODUCTION

1.1 Motivation

1.1.1 Finite elements in biomechanics

The finite element (FE) method is one of the most powerful numerical methods for solving the differential equations of physics. It has been used in many fields of engineering and physics, including solid mechanics, fluid mechanics, electromagnetism, heat transfer, and many more. The FE method has been used in the field of biomechanics as early as the 1970s (e.g. (Belytschko et al.; Davids and Mani; Doyle and Dobrin; Farah, Craig and Sikarskie; Janz and Grimm; Matthews and West)). Since that time, computational modeling in biomechanics and biophysics has become a standard methodology, both for understanding experimental outcomes, and as an investigative approach in its own right, and has facilitated advancements in nearly aspect of the field. Applications range from molecular dynamics, cell motility and mechanics, cardiovascular mechanics, musculoskeletal biomechanics, and tissue engineering. Advancements in high-performance computing hardware, as well as imaging techniques and geometry reconstruction, have facilitated high-resolution modeling of patient-specific geometries (e.g. (Anderson, Peters, et al.; Portnoy et al.; Schmid-Schonbein and Diller; Speelman et al.; Spilker et al.; Crawford, Rosenberg and Keaveny)). Furthermore, the development of sophisticated constitutive models (e.g. mixture theory (Li and Herzog; Li and Herzog)) has produced more realistic representations of biological materials.

Despite these advancements in finite element technologies, the field of computational biomechanics remains one of the most challenging areas of applications: Geometries are often complex and can undergo very large deformations (e.g. muscle deformation around shoulder and knee joints (Ellis, Drury, et al.; Weiss et al.)) and sometimes consist of thin structures (e.g. articular cartilage (Anderson, Ellis, et al.), fascia, and blood vessel walls (Weisbecker et al.)). The response of biological tissue is often nonlinear, anisotropic (due to the presence of aligned fiber families (Weiss, Maker and Govindjee)), composited (e.g. multiple fiber families embedded in a matrix (Hollingsworth and Wagner)), and time-dependent (i.e. viscoelastic (Lujan, Underwood, et al.) or poroelastic). The characterization of the response of biological tissue may require the solution of coupled physical systems (e.g. fluid motion, changing solute, and solute concentrations due to dynamical processes (Ateshian et al.)). Also the modeling of rigid bodies and their coupling to deformable structures is important to model the connection of tissues (e.g. tendons, ligaments) to stiff structures (e.g. bone). In the area of articular joint mechanics (an area of particular interest in this work), the geometries may come in contact and the contacting surfaces can slide across each other over relatively large distances (Henak et al.). Biological tissues often show evidence of residual strains (Chuong and Fung; Gardiner, Weiss and Rosenber) and the inclusion of these strains is important for the accurate modeling of the mechanical response (Rausch and Kuhl). Finally, as the field of biomechanics is a rapidly evolving field, it is highly desirable that a computing platform can be extended easily to include novel constitutive laws or solution algorithms. All of these challenges make the development of finite element software that targets computational biomechanics difficult.

The lack of FE software that is tailored to the needs of the field has hampered research progress, dissemination of research, and sharing of models and results. Commercial packages that are most commonly used by the biomechanics community, such as Abaqus (Dassault Systèmes) and Ansys (Ansys, Inc.), are not specifically geared towards biological applications. The closed nature of commercial codes makes them difficult to verify (Anderson, Ellis and Weiss; Henninger et al.) or to compare results across different codes since the implementation details are often unavailable. Commercial codes do not always offer mechanisms for adding new features easily. This makes it challenging for researchers to implement and test new computational methods or constitutive models.

The limitations of commercial codes have forced many researchers to develop custom FE packages for solving problems in the biomechanics domain. One of the earliest examples of a finite element code developed for applications in biomechanics was by Simon, Wu, et al., who developed a custom code for modeling intervertebral disc segments based on porous media theories. FE implementations of biphasic theory for cartilage were presented by Spilker and co-workers, for small strains (Spilker and Suh) and finite deformation of the solid matrix (Suh, Spilker and Holmes), using penalty methods to enforce incompressibility. They considered various refinements to their approach (Donzelli et al.), including hybrid, mixed-penalty, and velocity-pressure formulations (Almeida and Spilker; Vermilyea and Spilker) as well as tissue anisotropy (Spilker, Donzelli and Mow) and explicit modeling of interstitial fluid viscosity (Chan, Donzelli and Spilker). u-p (displacement-pressure) formulations of the biphasic mixture theory were presented by Oomens, van Campen and Grootenboer and Wayne, Woo and Kwan, and a biphasic formulation accommodating solid matrix viscoelasticity was developed by Ehlers and

Markert. Custom-written FE codes for modeling deformation, electrokinetics, transport, and swelling of charged hydrated tissues were developed by Laible et al.; Simon, Liable, et al., Levenston, Frank and Grodzinsky, Sun et al., and Frijns et al.. Although these custom-written codes implemented more advanced modeling and solution algorithms than were available in commercial packages, few of these codes were actually made available in the public domain. This made verification and reproducibility of results challenging or impossible. Many of these codes were also not sufficiently general, but were instead designed for solving very specific problems and did not support all features that are necessary for analyses in computational biomechanics and biophysics. In addition, to the best of our knowledge, at the time development of FEBio started (c. 2005), none of these custom-written codes offered extended documentation, support, or mechanisms for continued development.

1.1.2 FEBio

Due to the many limitations of commercial and custom-written FE packages, there existed a genuine need for a new FE tool: a tool that would be designed by and for the biomechanics community. FEBio was developed to fill this need. In order to accomplish its goals, it would be designed around three pillars. First, it would specifically target the biomechanics community by implementing features that are highly relevant in the field, such as, accurate constitutive relations for mechanical, transport, and electrokinetic properties of tissues and cells; the ability to easily model anisotropy and inhomogeneity; and the ability to prescribe boundary conditions and loading scenarios that would allow researchers to model the complex interactions between biological structures. Second, the source code would be made freely available so that researchers can implement new algorithms (e.g. new constitutive models). This would greatly reduce the need for new custom-written codes as researchers can leverage the FEBio code to implement new algorithms. Third, emphasis would be placed on thorough documentation, support, and outreach to the community. It was hoped that this would make it much easier than before for researchers to develop new ideas and share them with others.

The first version of FEBio was released in 2007 and its initial focus was on the development of highly needed functionality that targets applications in joint contact mechanics. This included, among other things, solution algorithms for solving the nonlinear equations of continuum mechanics, constitutive models valid for large deformation, explicit representation of rigid body kinematics, and sliding and tied contact formulations. An initial implementation of a coupled displacement-pressure solution algorithm for biphasic models was provided as well. The FEBio code was developed in C++. Although many finite element codes, both commercial as well as open-source (e.g. Abaqus, Nike3D, Feap, Code Aster) use the FORTRAN language, an object-oriented approach to programming is widely accepted as a superior programming model compared to procedural programming (Cary et al.). C++ was chosen since it is an object-oriented programming language and it has been shown that it performs comparable to the FORTRAN language (Veldhuizen and Jernigan). Also, the possibility of developing efficient finite element codes based on object-oriented programming strategies has been demonstrated (Patzak and Bittnar; Zimmerman et al.). In addition, by relying extensively on existing design patterns (Gamma et al.; Heng and Mackie) many challenges in designing a complex software library could be overcome easily.

After several years of development, the code had grown in many ways that greatly expanded its applicability. Features were added to model multiple solvents that can interact with the solid matrix. Contact formulations that can handle fluid flow across the contact interface were implemented. Even a heat transfer solver was developed. However, the implementation of these advanced features proved difficult and required a substantial development effort, which highlighted some of the flaws of the initial design. Additionally, dissemination of new additions by users was still difficult. New releases of FEBio may be incompatible with the user's changes, either preventing the user from upgrading or forcing him or her to merge the new features to the new version. Also, the implementation of these new features, even minute changes, also required the recompilation of the entire FEBio source code tree.

Around 2010, roughly at the start of this work, the decision was made to redesign the code entirely and create a framework that was more flexible and that allowed users to add new features with minimal impact to the code structure. A major component of this new framework was a plugin mechanism that would allow users to develop new features independently from the main source code development. Since plugins do not require the compilation of the entire FEBio source code, they greatly simplify development of new features and facilitate the dissemination of these new additions.

1.1.3 Tetrahedral formulations for contact mechanics

Advances in imaging and computational methods make it possible to create and analyze detailed subject-specific surface models of biomechanical structures from high-resolution image data (e.g. MRI, CT). Volumetric meshes can then be constructed from these surfaces using an appropriate tessellation algorithm. Meshing algorithms that generate linear tetrahedral (TET4) meshes are often preferred due to the fact that it is relatively straightforward to perform automatic meshing, local and adaptive refinement (Delaunay; Hubsch et al.; Johnson and MacLeod; Prakash and Ethier; Donzelli et al.), and automatic tetrahedral meshing techniques are extremely robust and fast (Delaunay; Lo; Lo; Lohner; Shephard and Georges; Wrazidlo et al.). However, when used in large deformation solid mechanics, linear tetrahedral elements are overly stiff and can “lock” in problems where nearly incompressible materials and/or acute bending are encountered (Hughes; Zienkiewicz and Taylor). In addition, since the displacement derivatives of the shape functions are constant, TET4 elements can only represent a constant strain state. This necessitates a very fine discretization, often requiring long solution times to obtain even marginally converged results. Because of these issues, other element formulations have been explored. Trilinear hexahedral elements (HEX8) have seen wide use for discretizing articular cartilage in joint contact analyses. Although these elements also have a tendency to lock in a standard displacement formulation, several techniques are available for improving the performance of these elements, such as reduced integration and multifield approaches (Simo and Taylor). On the other hand, creating hexahedral meshes for complex geometries can be very challenging and time consuming. Although algorithms for automatic hex mesh generation, such as plastering (Blacker and Meyers) and whisker weaving (Tautges, Blacker and Mitchell), have been reported, the approaches are not always robust and for this reason have not been widely used. This motivates the search for alternative tetrahedral formulations that are accurate and can take advantage of automatic meshing algorithms.

Initially, we set out to explore methods based on nodal averaging of the deformation gradient. Originally proposed by Bonet, Marriott and Hassan and Dohrmann et al. in the context of small displacement theory, this element was extended to large deformations, such as in Gee et al. and Puso and Solberg. The original formulation was shown to be unstable. This was later resolved by adding the virtual work that corresponds to the classical linear tetrahedral formulation to the total virtual work, multiplied by a user-controlled scaling factor (Puso and Solberg). Although this may stabilize the formulation, this also reintroduces locking unless the second term uses a different "softer" constitutive model as was done in Puso and Solberg. This can be avoided by applying the stabilization only to the isochoric part of the stress (Gee et al.). However, despite these improvements to nodally integrated tetrahedrons, we found that these elements do not perform well in contact analyses (Maas et al.). This motivated us to look for other tetrahedral formulations. Quadratic tetrahedral element formulations appeared to be promising alternatives to linear tetrahedral formulations. These higher-order formulations maintain the advantages of tetrahedral mesh generation but suffer less from the numerical issues that plague the TET4 element. In addition, the numerical techniques that have proven successful for HEX8 elements can also be applied to higher-order tetrahedral formulations. They also offer an advantage over HEX8 elements because they can represent curved boundaries more accurately since their edges and faces can deform. We will look specifically at 10-node (TET10) and 15-node (TET15) quadratic tetrahedral element formulations. Although 10-node tetrahedral elements have been investigated as alternatives to HEX8 elements in general 3D FE analysis (e.g., (Cifuentes and Kalbag; Tadepalli, Erdemir and Cavanagh; Weingarten)), few studies have looked at their performance in contact (Bunbar et al.; Hao et al.; Tadepalli, Erdemir and Cavanagh; Wan, Hoa and Wen; Yang and Spilker) and none of these studies investigated their application for simulation of articular contact analyses.

To our knowledge, the 15-node tetrahedron has never been used in computational biomechanics and it has seen very limited use in nonlinear computational solid mechanics at all (Danielson), although it has been used in the fluid mechanics community (Bertrand, Gadbois and Tanguy).

Stress recovery is an important aspect of modeling problems in solid mechanics. It is well known that the strains and stresses predicted by the finite element method, using a standard displacement-based approach, are discontinuous across the element boundaries and thus are a poor approximation of the correct solution. A posteriori stress recovery methods attempt to obtain a more accurate prediction of the stresses based on the finite element results. Accurate recovery of stresses is not only important for visualization and analyses, but also for other applications such as error indicators in adaptive mesh refinement strategies. Various methods have been proposed in the literature for stress recovery. See the review articles (Ainsworth and Oden; Grätsch and Bathe) and the references therein. One of the most widely used methods is the Super-convergent Patch Recovery method (SPR) (Zienkiewicz and Zhu). This method relies on the existence of so-called super-convergent sampling points inside a finite element where the error in the stress converges faster than that of the standard finite element approximation. By fitting a polynomial to these sampling points, a more accurate solution for the stresses can be computed. In this work, the SPR method will be used to recover the stresses for the higher order tetrahedral formulations considered here.

1.1.4 Prestrain in biological tissues

Experimental observations show that connective tissues such as ligaments, tendons, and skeletal muscle retract when excised from the body. This is seen as an indication of *in* *situ* strain, i.e. strain that exists *in vivo* in the absence of loading and is relieved when the tissue is removed from the body. *In situ* strains for ligaments in diarthrodial joints are in the range of 3-10% (Gardiner, Weiss and Rosenber; Woo et al.) and it has been shown that they contribute to the stability of joints (Ellis, Lujan, et al.; Lujan, Dalton, et al.). Residual strain (i.e. strain that exists in the tissue after it is excised from the body) is also observed in many tissues such as arteries (Chuong and Fung), mitral leaflets (Rausch and Kuhl), and myocardium (Guccione, McCulloch and Waldman; Omens and Fung; Wang et al.). Both *in situ* strains and residual strains can contribute significantly to the mechanical response of the system. Therefore, their inclusion is often necessary in computational models of biological tissues to obtain reasonable predictions of tissue mechanics. We will use the term “prestrain” as a collective term for both *in situ* strains and residual strains and denote prestress as the stress that is induced by this prestrain.

Several computational methods to include prestrains in the context of finite element (FE) modeling have been presented in the literature (Alastrue et al.; Balzani, Schroder and Gross; Dhaher, Kwon and Barry; Holzapfel and Ogden; Pierce et al.; Rausch and Kuhl). Most of these methods assume that a global stress-free state exists and is computable. From this stress-free state, the unloaded, prestressed state is recovered by one of several methods. In one approach, the deformation of the global stress-free configuration to the prestressed configuration is assumed (Rausch and Kuhl). Another approach makes use of the analytical solution from a geometrically simplified problem (Holzapfel and Ogden; Pierce et al.). A third approach solves a forward finite element problem that recovers the unloaded, prestressed state by applying judicially chosen boundary conditions (Balzani, Schroder and Gross). Although these methods have been useful in studying the effects of prestrains and prestresses in computational methods, they have several limitations. Most of those methods assume that a global stress-free state exists. This assumption appears to be motivated mostly by the limitations of existing FE packages. Prestrain can often only be applied by translating the model to a sequence of forward analyses where a judicial choice of boundary conditions emulates the effect of residual strains. This obviously limits the application of residual strains in general and in fact, the existence of a globally stress-free state has been contested. Indeed, if the residual stress field forms from growth and remodeling of the tissue, there is no reason to assume that a globally stress-free configuration exists. Consider for instance the case of arteries, where the opening angle experiment is believed to relieve all the residual stresses (Fung and Liu) using a single radial cut. However, it was later shown that a single cut does not relieve all the residual stresses (Greenwald et al.; Vossoughi, Hedzaji and Borris) and multiple cuts are necessary to relieve the residual stresses. Other researchers have implemented the effects of residual strains directly into the constitutive models, but in this case, the constitutive assumptions are hardcoded in the implementation and the application to other constitutive models is not straight-forward. The most general method known to date for incorporating prestrain is the "General Prestressing Algorithm" (GPA) (Weisbecker, Pierce and Holzapfel). Although the method was presented in a different context, it can also be used to apply prestrain to a computational model (Pierce et al.). This method does not necessarily require a stress-free reference.

As noted before, the lack of a tailored environment for solving computational biomechanical problems has forced many researchers to implement the algorithms for applying prestrains into custom-developed codes; a practice that has hampered dissemination and complicated reproducibility of findings. Since FEBio's goal is to provide such a tailored environment, this work will expand FEBio's framework such that methods for incorporating prestrain can be implemented easily. The framework will focus on methods that do not require a stress-free reference configuration and will be general enough so that it can be applied to a wide range of constitutive models. As an example of an application of the framework, methods will be explored that apply a given prestrain field to a model, a question of particular importance when the prestrain data are obtained from experimental data. Inaccuracies in the experimental prestrain data may render the applied prestrain field incompatible with the reference geometry. This will result in an additional, nonphysical distortion of the geometry that may affect the accuracy of the outcome. For such applications, the ability to quantify this distortion, and if possible, to correct the distortion may be critical for obtaining meaningful results.

1.2 Research Goals

The overall objectives of this research were to redesign the FEBio code structure in a way that would allow users to add new features in an easy and convenient way. Then, using this improved structure, two common, yet important aspects of simulating biological structures would be addressed: using reliable tetrahedral element formulations for solving large deformation contact problems; and applying prestrain to models of biological tissues.

1.2.1 Aim 1

Aim 1 is to implement a general framework for extending FEBio that allows new features to be added without the need to make any changes to the source code. This framework will also provide the ability to create plugins, which allows users to add new functionality independently from FEBio's development.

1.2.2 Aim 2

Aim 2 is to determine effective tetrahedral element formulations for joint contact mechanics. Specifically, the robustness in terms of convergence behavior in nonlinear solution strategies of several quadratic tetrahedral formulations and several integration strategies will be explored. Stresses will be recovered using a Super-convergent Patch Recovery method and the accuracy of these stresses, especially the predicted contact stresses, will be compared for the various formulations.

1.2.3 Aim 3

Aim 3 is to develop a novel computational framework for including residual strains based on methodology that does not assume the existence or computability of a globally stress-free reference configuration. The approach will be applicable to a wide range of constitutive models and prestrain applications.

1.3 Summary of Chapters

In Chapter 2, the theoretical framework is presented on which FEBio is based. The governing equations of nonlinear mechanics are reviewed. Details on the specific constitutive models, contact formulations, and rigid body algorithms are presented as well. This chapter also discusses some of the implementation aspects and explains how FEBio interacts with third-party linear solvers. This chapter also presents an extensive list of verification problems that show that FEBio not only can reproduce accurate results, but also that it is competitive in terms of performance, compared to other FE codes used in the field.

The discussion of the implementation aspects behind FEBio continues in Chapter 3. The focus here is on the modifications to the original design that greatly simplify the implementation of new features. This chapter also presents the plugin framework that allows users to make additions to the code, without the need to modify or build the FEBio source code directly. The chapter ends with several examples of plugins that illustrate the versatility of the new framework.

The next three chapters tackle some of the challenges in modeling joint contact mechanics. In Chapter 4, quadratic tetrahedral formulations are explored as alternatives to the “gold-standard” linear hexahedral elements. Two particular quadratic tetrahedrons are compared to linear tetrahedral and linear hexahedral elements. Several integration rules are investigated for their accuracy. This chapter also discusses stress-recovery of results obtained with quadratic tetrahedral elements. In Chapter 5, additional simulations are presented that further strengthen the observations made in the previous chapter. Chapter 6 presents a computational framework for modeling prestrain in FE models. Several applications are discussed, including enforcing fiber stretch onto a model, how to obtain a prestressed state when the prestressed configuration is known, and how to use the framework to solve certain types of inverse FE problems.

Chapter 7 presents an overall summary of the work and discusses the significance of this work. It also outlines some future directions that have yet to be explored. Finally, Chapter 8 presents an overview of additional projects that were led by the author that have contributed in a significant way to the development of the FEBio software project.

1.4 References

Ansys, Inc. *Ansys*. Vers.

Ainsworth, Mark, and J. Tinsley Oden. "A Posteriori Error Estimation in Finite Element Analysis." *Comput. Methods Appl. Mech. Engrg* 142.1 (1997): 1-88.

Alastrue, V., et al. "Assessing the Use of the "Opening Angle Method" to Enforce Residual Stresses in Patient-Specific Arteries." *Ann. Biomed. Eng.* 35.10 (2007): 1821-37.

Almeida, E. S., and R. L. Spilker. "Mixed and Penalty Finite Element Models for the Nonlinear Behavior of Biphasic Soft Tissues in Finite Deformation: Part I - Alternate Formulations." *Comput. Methods Biomech. Biomed. Eng.* 1.1 (1997): 25-46.

Anderson, A. E., et al. "Validation of Finite Element Predictions of Cartilage Contact Pressure in the Human Hip Joint." *J. Biomech. Eng.* 130.5 (2008): 1008-18.

Anderson, A. E., B. J. Ellis, and J. A. Weiss. "Verification, Validation and Sensitivity Studies in Computational Biomechanics." *Comput. Methods Biomech. Biomed. Eng.* 10.3 (2007).

Anderson, A. E., et al. "A Subject-Specific Finite Element Model of the Pelvis: Development, Validation and Sensitivity Studies." *J. Biomech. Eng.* 127.3 (2005): 364-73.

Ateshian, G. A., et al. "Computational Modeling of Chemical Reactions and Interstitial Growth and Remodeling Involving Charged Solutes and Solid-Bound Molecules." *Biomech. Model. Mechanobiol* 13.5 (2014): 1105-20.

Balzani, D., J. Schroder, and D. Gross. "Numerical Simulation of Residual Stresses in Arterial Walls." *Comp. Mater. Sci.* 39 (2007): 117-23.

---. "Simulation of Discontinuous Damage Incorporating Residual Stresses in Circumferentially Overstretched Atherosclerotic Arteries." *Acta Biomater.* 2.6 (2006): 609-18.

Belytschko, T., et al. "Finite Element Stress Analysis of an Intervertebral Disc." *J. Biomech.* 7.3 (1974): 277-85.

Bertrand, F.H., M.R. Gadbois, and P.A. Tanguy. "Tetrahedral Elements for Fluid Flow." *Intl. J. Num. Meth. Eng.* 33.6 (1992): 1251-67.

Blacker, Ted D., and Ray J. Meyers. "Seams and Wedges in Plastering: A 3-D Hexahedral Mesh Generation Algorithm." *Eng. Comput.* 9.2 (1993): 83-93.

Bonet, J., H. Marriott, and O. Hassan. "An Averaged Nodal Deformation Gradient Linear Tetrahedral Element for Large Strain Explicit Dynamic Applications." *Commun. Numer. Math. En.* 17.8 August (2001): 551-61.

Bunbar, W.L., et al. "An Evaluation of Three-Dimensional Diarthrodial Joint Contact Using Penetration Data and the Finite Element Method." *J. Biomech. Eng.* 123.4 (2001): 333-40.

Cary, J.R., et al. "Comparison of C++ and Fortran 90 for Object-Oriented Scientific Programming." *Comput. Phys. Commun.* 105.1 (1998): 20-36.

Chan, B., P. S. Donzelli, and R. L. Spilker. "A Mixed-Penalty Biphasic Finite Element Formulation Incorporating Viscous Fluids and Material Interfaces." *Ann. Biomed. Eng.* 28.6 (2000): 589-97.

Chuong, C. J., and Y. C. Fung. "On Residual Stresses in Arteries." *J. Biomech. Eng.* 108.2 (1986): 189-92.

Cifuentes, A.O., and A. Kalbag. "A Performance Study of Tetrahedral and Hexahedral Elements in 3-D Finite Element Structural Analysis." *Finite Elem. Anal. Des.* 12.3-4 (1992): 313-18.

Crawford, R. P., W. S. Rosenberg, and T. M. Keaveny. "Quantitative Computed Tomography-Based Finite Element Models of the Human Lumbar Vertebral Body: Effect of Element Size on Stiffness, Damage, and Fracture Strength Predictions." *J. Biomech. Eng.* 125.4 (2003): 434-8.

Danielson, Kent T. "Fifteen Node Tetrahedral Elements for Explicit Methods in Nonlinear Solid Dynamics." *Comput. Method. Appl. M.* 272 (2014): 160-80.

Davids, N., and M. K. Mani. "A Finite Element Analysis of Endothelial Shear Stress for Pulsatile Blood Flow." *Biorheology* 11.2 (1974): 137-47.

Delaunay, B. N. (1934) "Sur la Sphere" Vide. Izvestia Akademia Nauk SSSR, VII Seria, Otdelenie Matematicheskii i Estestvennyka Nauk Vol 7 pp.793-800. ""Sur La Sphere" Vide." *Izvestia Akademia Nauk SSSR, VII Seria, Otdelenie Matematicheskii i Estestvennyka Nauk* 7 (1934): 793-800.

Dhaher, Y. Y., T. H. Kwon, and M. Barry. "The Effect of Connective Tissue Material Uncertainties on Knee Joint Mechanics under Isolated Loading Conditions." *J. Biomech.* 43.16 (2010): 3118-25.

Dohrmann, C.R., et al. "Node-Based Uniform Strain Elements for Three-Node Triangular and Four-Node Tetrahedral Meshes." *Intl. J. Num. Meth. Eng.* 47.9 (2000): 1549-68.

Donzelli, P. S., et al. "Automated Adaptive Analysis of the Biphasic Equations for Soft Tissue Mechanics Using a Posteriori Error Indicators." *Intl. J. Num. Meth. Eng.* 34.3 (1992): 1015-33.

Doyle, J. M., and P. B. Dobrin. "Finite Deformation Analysis of the Relaxed and Contracted Dog Carotid Artery." *Microvasc. Res.* 3.4 (1971): 400-15.

Ehlers, W., and B. Markert. "A Linear Viscoelastic Biphasic Model for Soft Tissues Based on the Theory of Porous Media." *J. Biomech. Eng.* 123.5 (2001): 418-24.

Ellis, B. J., et al. "Finite Element Modeling of the Glenohumeral Capsule Can Help Assess the Tested Region During Clinical Exam." *Comput. Method. Biomec.* 13.3 (2010): 413-18.

Ellis, B. J., et al. "Medial Collateral Ligament Insertion Site and Contact Forces in the Acl-Deficient Knee." *J. Orthop. Res.* 24.4 (2006): 800-10.

Farah, J. W., R. G. Craig, and D. L. Sikarskie. "Photoelastic and Finite Element Stress Analysis of a Restored Axisymmetric First Molar." *J. Biomech.* 6.5 (1973): 511-20.

Frijns, A. J., et al. "Numerical Simulation of Deformations and Electrical Potentials in a Cartilage Substitute." *Biorheology* 40.1-3 (2003): 123-31.

Fung, Y.C., and S.Q. Liu. "Change of Residual Strains in Arteries Due to Hypertrophy Caused by Aortic Constriction." *Circ. Res.* 65 (1989): 1340-49.

Gamma, Erich, et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995. Print.

Gardiner, J.C., J.A. Weiss, and T.D. Rosenber. "Strain in the Human Medial Collateral Ligament During Valgus Loading of the Knee." *Clin. Orthop. Relat. Res.*391 (2001): 266-74.

Gee, M.W., et al. "A Uniform Nodal Strain Tetrahedron with Isochoric Stabilization." *Intl. J. Num. Meth. Eng.* 78 (2009): 429-43.

Grätsch, Thomas, and Klaus-Jürgen Bathe. "A Posteriori Error Estimation Techniques in Practical Finite Element Analysis." *Comput. Struct.* 83.4–5 (2005): 235-65.

Greenwald, S.E., et al. "Experimental Investigation of the Distribution of Residual Strains in the Artery Wall." *J. Biomech. Eng.* 1994.4 (1997): 438-44.

Guccione, J. M., A. D. McCulloch, and L. K. Waldman. "Passive Material Properties of Intact Ventricular Myocardium Determined from a Cylindrical Model." *J. Biomech. Eng.* 113.1 (1991): 42-55.

Hao, Z, et al. "The Effect of Boundary Condition on the Biomechanics of a Human Pelvic Joint under an Axial Compressive Load: A Three-Dimensional Finite Element Model." *J. Biomech. Eng.* 133.10 (2011).

Henak, C.L., et al. "Specimen-Specific Predictions of Contact Stress under Physiological Loading in the Human Hip: Validation and Sensitivity Studies." *Biomech. Model. Mechanobiol* 13.2 (2014): 387-400.

Heng, B. C. P., and R. I. Mackie. "Using Design Patterns in Object-Oriented Finite Element Programming." *Comput. Struct.* 87.15–16 (2009): 952-61.

Henninger, H. B., et al. "Validation of Computational Models in Biomechanics." *Proc. Inst. Mech. Eng. H* 224.7 (2010): 801-12.

Hollingsworth, N.T., and D.R. Wagner. "Modeling Shear Behavior of the Annulus Fibrosus." *J. Mech. Behav. Biomed. Mater.* 4.7 (2011): 1103-14.

Holzapfel, G. A., and R. W. Ogden. "Modelling the Layer-Specific Three-Dimensional Residual Stresses in Arteries, with an Application to the Human Aorta." *J. R. Soc. Interface* 7.46 (2010): 787-99.

Hubsch, P. F., et al. "Adaptive Finite-Element Approach for Analysis of Bone/Prosthesis Interaction." *Med. Biol. Eng. Comput.* 33.1 (1995): 33-7.

Hughes, T.J.R. *The Finite Element Method*. Dover, 2000. Print.

Janz, R. F., and A. F. Grimm. "Finite-Element Model for the Mechanical Behavior of the Left Ventricle. Prediction of Deformation in the Potassium-Arrested Rat Heart." *Circ. Res.* 30.2 (1972): 244-52.

Johnson, C. R., and R. S. MacLeod. "Adaptive Local Regularization Methods for the Inverse Ecg Problem." *Prog. Biophys. Mol. Biol.* 69.2-3 (1998): 405-23.

Laible, J. P., et al. "A Poroelastic-Swelling Finite Element Model with Application to the Intervertebral Disc." *Spine* 18.5 (1993): 659-70.

Levenston, M. E., E. H. Frank, and A. J. Grodzinsky. "Electrokinetic and Poroelastic Coupling During Finite Deformations of Charged Porous Media." *J. Appl. Mech.* 66.2 (1999): 323-33.

Li, L. P., and W. Herzog. "Arthroscopic Evaluation of Cartilage Degeneration Using Indentation Testing--Influence of Indenter Geometry." *Clin. Biomech.* 21.4 (2006): 420-6.

---. "Electromechanical Response of Articular Cartilage in Indentation--Considerations on the Determination of Cartilage Properties During Arthroscopy." *Comput. Methods*

Biomech. Biomed. Eng. 8.2 (2005): 83-91.

Lo, S. H. "Volume Discretization into Tetrahedra-I. Verification and Orientation of Boundary Surfaces." *Computers and Structures* 39.5 (1991): 493-500.

---. "Volume Discretization into Tetrahedra - Ii. 3d Triangulation by Advancing Front Approach." *Computers and Structures* 39.5 (1991): 501-11.

Lohner, R. "Progress in Grid Generation Via the Advancing Front Technique." *Engineering with Computers* 12 (1996): 186-210.

Lujan, T. J., et al. "Effect of Acl Deficiency on Mcl Strains and Joint Kinematics." *J. Biomech. Eng.* 129.3 (2007): 386-92.

Lujan, T. J., et al. "Contribution of Glycosaminoglycans to Viscoelastic Tensile Behavior of Human Ligament." *J. Appl. Physiol.* 106.2 (2008): 423-31.

Maas, S.A., et al. "Implementation and Verification of a Nodally-Integrated Tetrahedral Element in Febio." *SCI Technical Report*: University of Utah, 2011. Print.

Matthews, F. L., and J. B. West. "Finite Element Displacement Analysis of a Lung." *J. Biomech.* 5.6 (1972): 591-600.

Omens, J. H., and Y. C. Fung. "Residual Strain in Rat Left Ventricle." *Circ. Res.* 66.1 (1990): 37-45.

Oomens, C. W., D. H. van Campen, and H. J. Grootenboer. "A Mixture Approach to the Mechanics of Skin." *J. Biomech.* 20.9 (1987): 877-85.

Patzak, B., and Z. Bittnar. "Design of Object Oriented Finite Element Code." *Adv. Eng. Softw.* 32 (2001): 759-67.

Pierce, D. M., et al. "A Method for Incorporating Three-Dimensional Residual Stretches/Stresses into Patient-Specific Finite Element Simulations of Arteries." *J. Mech. Behav. Biomed.* 47 (2015): 147-64.

Portnoy, S., et al. "Real-Time Patient-Specific Finite Element Analysis of Internal Stresses in the Soft Tissues of a Residual Limb: A New Tool for Prosthetic Fitting." *Ann. Biomed. Eng.* 35.1 (2007): 120-35.

Prakash, S., and C. R. Ethier. "Requirements for Mesh Resolution in 3d Computational Hemodynamics." *J. Biomech. Eng.* 123.2 (2001): 134-44.

Puso, M. A., and J. Solberg. "A Stabilized Nodally Integrated Tetrahedral." *Intl. J. Num. Meth. Eng.* 67.6 (2006): 841-67.

Rausch, M. K., and E. Kuhl. "On the Effect of Prestrain and Residual Stress in Thin Biological Membranes." *J. Mech. Phys. Solids* 61.9 (2013): 1955-69.

Schmid-Schonbein, G. W., and K. R. Diller. "Transport Processes in Biomedical Systems: A Roadmap for Future Research Directions." *Ann. Biomed. Eng.* 33.9 (2005): 1136-41.

Shephard, M. S., and M. K. Georges. "Three-Dimensional Mesh Generation by Finite Octree Technique." *International Journal for Numerical Methods in Engineering* 32 (1991): 709-49.

Simo, J.C. , and R.L. Taylor. "Quasi-Incompressible Finite Elasticity in Principal Stretches: Continuum Basis and Numerical Algorithms." *Comput. Methods Appl. Mech. Engrg* 85 (1991): 273-310.

Simon, B. R., et al. "A Poroelastic Finite Element Formulation Including Transport and Swelling in Soft Tissue Structures." *J. Biomech. Eng.* 118.1 (1996): 1-9.

Simon, B. R., et al. "Poroelastic Dynamic Structural Models of Rhesus Spinal Motion Segments." *Spine* 10.6 (1985): 494-507.

Speelman, L., et al. "Effects of Wall Calcifications in Patient-Specific Wall Stress Analyses of Abdominal Aortic Aneurysms." *J. Biomech. Eng.* 129.1 (2007): 105-9.

Spilker, R. L., P. S. Donzelli, and V. C. Mow. "A Transversely Isotropic Biphasic Finite Element Model of the Meniscus." *J. Biomech.* 25.9 (1992): 1027-45.

Spilker, R. L., et al. "Morphometry-Based Impedance Boundary Conditions for Patient-Specific Modeling of Blood Flow in Pulmonary Arteries." *Ann. Biomed. Eng.* 35.4 (2007): 546-59.

Spilker, R. L., and J. K. Suh. "Formulation and Evaluation of a Finite Element Model for the Biphasic Model of Hydrated Soft Tissues." *Comput. Struct.* 35.4 (1990): 425-39.

Suh, J. K., R. L. Spilker, and M. H. Holmes. "A Penalty Finite Element Analysis for Nonlinear Mechanics of Biphasic Hydrated Soft Tissue under Large Deformation." *Intl. J. Num. Meth. Eng.* 32.7 (1991): 1411-39.

Sun, D. N., et al. "A Mixed Finite Element Formulation of Triphasic Mechano-Electrochemical Theory for Charged, Hydrated Biological Soft Tissues." *Intl. J. Num. Meth. Eng.* 45.10 (1999): 1375-402.

Tadepalli, S.C., A. Erdemir, and P.R. Cavanagh. "Comparison of Hexahedral and Tetrahedral Elements in Finite Element Analysis of the Foot and Footwear." *J. Biomech.* 44 (2011): 2337-43.

Tautges, Timothy J., Ted Blacker, and Scott A. Mitchell. "Whisker Weaving Algorithm: A Connectivity-Based Method for Constructing All-Hexahedral Finite Element Meshes." *Intl. J. Num. Meth. Eng.* 39.19 Oct 15 (1996): 3327-49.

Veldhuizen, Todd L., and M. Ed Jernigan. "Will C++ Be Faster Than Fortran?" *Scientific Computing in Object-Oriented Parallel Environments*. Eds. Ishikawa, Yutaka, et al. Vol. 1343. Lecture Notes in Computer Science: Springer Berlin Heidelberg, 1997. 49-56. Print.

Vermilyea, M. E., and R. L. Spilker. "Hybrid and Mixed-Penalty Finite Elements for 3-D Analysis of Soft Hydrated Tissue." *Intl. J. Num. Meth. Eng.* 36.24 (1993): 4223-43.

Vossoughi, J., Z. Hedzaji, and F.S. Borris. "Intimal Residual Stress and Strain in Large Arteries." *Bioengineering Conference ASME*. 1993. Print.

Wan, C., Z Hoa, and S Wen. "The Effect of the Variation in Acl Constitutive Model on Joint Kinematics and Biomechanics under Different Loads: A Finite Element Study." *J. Biomech. Eng.* 135.4 (2013): 041002.

Wang, H. M., et al. "A Modified Holzapfel-Ogden Law for a Residually Stressed Finite Strain Model of the Human Left Ventricle in Diastole." *Biomech. Model. Mechanobiol.* 13.1 (2014): 99-113.

Wayne, J. S., S. L. Woo, and M. K. Kwan. "Application of the U-P Finite Element Method to the Study of Articular Cartilage." *J. Biomech. Eng.* 113.4 (1991): 397-403.

Weingarten, V.I. "The Controversy over Hex or Tet Meshing." *Mach. Des.* 66.8 (1994): 74-78.

Weisbecker, H., D. M. Pierce, and G. A. Holzapfel. "A Generalized Prestressing Algorithm for Finite Element Simulations of Preloaded Geometries with Application to the Aorta." *Int. J. Numer. Methods Biomed.* 30 (2014): 857-72.

Weisbecker, H., et al. "Layer-Specific Damage Experiments and Modeling of Human Thoracic and Abdominal Aortas with Non-Athersosclerotic Intimal Thickening." *J. Mech. Behav. Biomed. Mater.* 2012.12 (2012): 93-106.

Weiss, J. A., et al. "Three-Dimensional Finite Element Modeling of Ligaments: Technical Aspects." *Med. Eng. Phys.* 27.10 (2005): 845-61.

Weiss, J.A., B.N. Maker, and S. Govindjee. "Finite Element Implementation of Incompressible, Transversely Isotropic Hyperelasticity." *Comput. Method. Appl. M.* 135 (1996): 107-28.

Woo, S. L., et al. "Measurement of Changes in Ligament Tension with Knee Motion and Skeletal Maturation." *J. Biomech. Eng.* 112.1 (1990): 46-51.

Wrazidlo, W., et al. "An Alternative Method of Three-Dimensional Reconstruction from Two- Dimensional Ct and Mr Data Sets." *Eur J Radiol* 12.1 (1991): 11-6.

Yang, T., and R. L. Spilker. "A Lagrange Multiplier Mixed Finite Element Formulation for Three-Dimensional Contact of Biphasic Tissues." *J. Biomech. Eng.* 129.3 (2006): 457-71.

Zienkiewicz, O. C. , and R. L. Taylor. *The Finite Element Method, Volume 1: Basic Formulation and Linear Problems*. London1989. Print.

Zienkiewicz, O.C., and J.Z. Zhu. "The Superconvergent Patch Recovery and a Posteriori Error Estimates. Part 1: The Recovery Technique." *Intl. J. Num. Meth. Eng.* 33 (1992): 1331-634.

Zimmerman, Th., et al. "Aspects of an Object-Oriented Finite Element Environement." *Comput. Struct.* 68 (1998).

CHAPTER 2[1]

FEBIO: FINITE ELEMENTS FOR BIOMECHANICS

2.1 Abstract

In the field of computational biomechanics, investigators have primarily used commercial software that is neither geared toward biological applications nor sufficiently flexible to follow the latest developments in the field. This lack of a tailored software environment has hampered research progress, as well as dissemination of models and results. To address these issues, we developed the FEBio software suite (http://mrl.sci.utah.edu/software/febio), a nonlinear implicit finite element (FE) framework, designed specifically for analysis in computational solid biomechanics. This paper provides an overview of the theoretical basis of FEBio and its main features. FEBio offers modeling scenarios, constitutive models, and boundary conditions, which are relevant to numerous applications in biomechanics. The open-source FEBio software is written in C++, with particular attention to scalar and parallel performance on modern computer architectures. Software verification is a large part of the development and maintenance of FEBio, and to demonstrate the general approach, the description and results of several problems from the FEBio Verification Suite are presented and compared to analytical solutions or results from other established and verified FE codes. An additional

[1] Reprinted from *Journal of Biomechanical Engineering*, Volume 134, Issue 1, Maas SA, Ellis BJ, Ateshian GA, Weiss JA, “FEBio: Finite Elements for Biomechanics”, 2012, with permission from ASME.

simulation is described that illustrates the application of FEBio to a research problem in biomechanics. Together with the pre- and postprocessing software PreView and Postview, FEBio provides a tailored solution for research and development in computational biomechanics.

2.2 Introduction

Accurate, quantitative simulations of the biomechanics of living systems and their surrounding environment have the potential to facilitate advancements in nearly every aspect of medicine and biology. For instance, computational models can yield estimates of stress and strain data over the entire continuum of interest, which is especially advantageous for locations where it may be difficult or impossible to obtain experimental measurements. Computational modeling in biomechanics has already become a standard methodology, both for interpreting the biomechanical and biophysical basis of experimental results and as an investigative approach in its own right when experimental investigation is difficult or impossible. Applications span all fields of the biomedical sciences, including areas as diverse as molecular dynamics, cell motility and mechanics, cardiovascular mechanics, musculoskeletal biomechanics and tissue engineering. Advancements in imaging techniques and geometry reconstruction have opened the door to patient-specific modeling [1-6], which could revolutionize the way clinicians diagnose and treat certain pathologies. Continuing improvements in speed and availability of high performance computing hardware have allowed use of finely discretized geometries (e.g. high resolution representations of vertebral bodies [7]) and sophisticated constitutive models (e.g. mixture theory [8,9]), with the hope that these added complexities will produce more realistic representations of biological materials.

The finite element (FE) method is by far the most common numerical discretization and solution technique that has been used in computational biosolid mechanics. The FE method provides a systematic approach for assembling the response of a complex system from individual contributions of elements, and thus it is ideal for the complex geometries often encountered in biomechanical systems. It also provides a consistent way to address material inhomogeneities and differences in constitutive models between disjoint or continuous parts of a model. The solution procedure involves the consideration of overall energy minimization and/or other fundamental physical balance laws to determine unknown field variables over the domain. The FE method has been applied to problems in biomechanics as early as the 1970s (see, e.g., Refs. [10-15]). The application of finite element analysis in biomechanics research and design has increased exponentially over the last 30 years as commercial software availability has improved and researchers obtained better access to appropriate computing platforms. Applications have spanned from the molecular to cellular, tissue, and organ levels.

However, the lack of a FE software environment that is tailored to the needs of the field has hampered research progress, dissemination of research, and sharing of models and results. Investigators have primarily used commercial software, but these packages are not specifically geared toward biological applications, are difficult to verify [16,17], preclude the easy addition and sharing of new features such as constitutive models, and are not sufficiently general to encompass the broad framework needed in biomechanics. To address these issues, we developed FEBio (an acronym for “Finite Elements for Biomechanics”), a nonlinear implicit finite element framework designed specifically for analysis in computational solid biomechanics [18].

Arguably, the most important aspect of developing a new FE code is proper verification. The American Society of Mechanical Engineer's 'Guide for Verification and Validation in Computational Solid Mechanics' [19] defines verification as:

> The process of determining that a computational model accurately represents the underlying mathematical model and its solution. In essence, verification is the process of gathering evidence to establish that the computational implementation of the mathematical model and its associated solution are correct.

In the case of computational solid biomechanics, the mathematical model is based on the governing equations of continuum mechanics (in particular the conservation of linear momentum), the associated boundary conditions, initial conditions, and constitutive equations. Development of a numerical method of analysis based on the mathematical model requires numerical discretization, solution algorithms, and convergence criteria [19,20]. To verify the numerical methods and computational implementation of the mathematical model in FEBio, it must be demonstrated that it gives the correct solution to a set of benchmark problems that consists of either analytical solutions or results from established FE codes.

Our long term goal is to develop a freely available, extensible finite element modeling framework for solid mechanics, fluid mechanics, solute transport, and electrokinetics in biological cells, tissues, and organs, based around the FEBio framework. To date, no such tools are available for general use in the public domain. The specific objectives of this paper are to introduce the theoretical framework of FEBio and to present results for a suite of verification problems that simultaneously illustrates some of its capabilities.

2.3 Overview

FEBio offers modeling scenarios, constitutive models and boundary conditions that are relevant to numerous applications in biomechanics. FEBio supports both quasi-static and dynamic analyses. In a quasi-static analysis, the equilibrium response of the system is sought and inertial terms are ignored. An incremental iterative solution is obtained by discretizing applied loads and other boundary condition in quasi-time. For deformable porous media, a coupled solid-fluid problem is solved. In a dynamic analysis, the inertial terms are included to calculate the time-dependent response of the system. FEBio uses an implicit time integration scheme for both types of analyses.

Many different constitutive models are available to represent biological materials and synthetic biomaterials. Many of the materials are based on the framework of hyperelasticity. Both isotropic and anisotropic constitutive models are available. FEBio also contains a *rigid body* constitutive model. This model can be used to represent materials or structures whose deformation is negligible compared to that of other materials in the overall model.

FEBio supports a wide range of boundary conditions to model biological interactions. These include prescribed displacements, nodal forces, surface tractions, pressure forces, and body forces (e.g. gravity). For dynamic problems, initial conditions are available for prescribing the initial values for velocity, acceleration and, in the case of biphasic analysis, fluid pressure and fluid flux. Deformable models can be connected to rigid bodies. This allows the user to model prescribed rotations and torques for rigid segments, thereby allowing the coupling of rigid body mechanics with deformable continuum mechanics. FEBio offers several ways to represent contact between rigid and/or deformable materials. Using a sliding surface, the user can connect two surfaces that are allowed to separate and slide across each other but are not allowed to penetrate. If the contacting surfaces are poroelastic, the user can choose to allow fluid to flow through the contact interface in the presence of a fluid pressure gradient across the interface. The tied interface can be used to tie two possibly nonconforming surfaces together.

FEBio is open source, and both the executables and source code may be downloaded free of charge (http://mrl.sci.utah.edu/software/febio). Each downloadable package contains the executable, verification and example problems, a user's manual and a theory manual. Precompiled executables are available for the Windows 7, Windows XP, Linux 64 bit, Linux 32 bit, and Mac OSX platforms. FEBio can be an excellent starting point for researchers who wish to implement new technologies. Although adding new functionality to FEBio requires some C++ programming skills, the modular structure of the code greatly facilitates this process. The source code is commented clearly and HTML documentation exists as well. To facilitate support, bug tracking and feature requests, a public forum has been created (http://mrl.sci.utah.edu/forums).

Two software packages that support the use of the FEBio software have also been created. PreView is a finite element preprocessor that offers a graphical user interface that facilitates the process of defining finite element models. The user can import geometry, assign material parameters, define boundary and contact conditions, and export the final model as a FEBio input file. The results of a FEBio run can subsequently be analyzed and visualized in the finite element postprocessor PostView. This environment offers tools to inspect and analyze the model results similar to those found in most commercial finite element software packages. Both PreView and PostView may be downloaded for free from the same software website.

2.4 Theory and Implementation

The following section assumes knowledge of nonlinear continuum mechanics and finite element methods. The FEBio Theory Manual [21] and the references below can be consulted for more detailed explanations of the presented theory.

2.4.1 Weak form, linearization and finite element discretization

Generally, a finite element formulation is established by a variational statement, which represents the weak form of a physical law. In mechanics, this can either be written as the minimization of an energy function or, alternatively, the virtual work equation can serve as the starting point [22]. The spatial form of the virtual work equation, representing the weak form of conservation of linear momentum for a deformable body, can be written:

$$\delta W = \int_{v} \boldsymbol{\sigma} : \delta \mathbf{d} dv - \int_{v} \mathbf{f} \cdot \delta \mathbf{v} dv - \int_{\partial v} \mathbf{t} \cdot \delta \mathbf{v} da = 0 . \tag{2.1}$$

Here, $\boldsymbol{\sigma}$ is the 2nd order Cauchy stress tensor, $\delta \mathbf{d}$ is the 2nd order virtual rate of deformation tensor, $\delta \mathbf{v}$ is a virtual velocity, and V and ∂v represent the volume and surface in the deformed configuration, respectively. Equation (2.1) is a highly nonlinear function of the deformation, and an iterative method is necessary to solve for the deformation. In FEBio, this equation is solved using an incremental-iterative strategy based on Newton's method, which requires the linearization of Eq. (2.1) [23].

The directional derivative of Eq. (2.1) is needed for the linearization. In an iterative procedure, the deformation $\boldsymbol{\varphi}$ will be approximated by a trial solution $\boldsymbol{\varphi}_k$. Linearization of the virtual work equation around this trial state gives:

$$\delta W\left(\boldsymbol{\varphi}_k,\delta\mathbf{v}\right)+D\delta W\left(\boldsymbol{\varphi}_k,\delta\mathbf{v}\right)\cdot\mathbf{u}=0\,, \tag{2.2}$$

where $D\delta W\left(\boldsymbol{\varphi}_k,\delta\mathbf{v}\right)\cdot\mathbf{u}$ is the directional derivative of the virtual work in the direction **u**.

In the finite element method, the domain is divided into connected subunits called finite elements. The discretization process is established by interpolating the position **X** of a point within a finite element in terms of the coordinates $\mathbf{X}_a$ of the nodes *a* that define the finite element and the shape functions, *N*$_a$:

$$\mathbf{X}=\sum_{a=1}^{n}N_a\mathbf{X}_a\,, \tag{2.3}$$

where *n* is the number of nodes in a finite element. Quantities such as spatial coordinates, displacement, velocity, and their virtual equivalents can be interpolated similarly.

The discretization of Eq. (2.2) with Eq. (2.3) leads to the discrete form of the nonlinear finite element equations [24]:

$$\delta\mathbf{v}^T\cdot\mathbf{K}\cdot\mathbf{u}=-\delta\mathbf{v}^T\cdot\mathbf{R}\ \ . \tag{2.4}$$

Here, **K** is the stiffness matrix, which is mostly defined by the constitutive model, **u** is the vector of nodal displacements, and **R** is the residual vector, which measures the difference between internal and external forces. Since the virtual velocities $\delta\mathbf{v}$ are arbitrary, they can be eliminated. An iterative solution scheme based on Newton’s method can be formulated as follows:

$$\mathbf{K}\left(\mathbf{x}_k\right)\cdot\mathbf{u}=-\mathbf{R}\left(\mathbf{x}_k\right);\quad \mathbf{x}_{k+1}=\mathbf{x}_k+\mathbf{u}\,. \tag{2.5}$$

Here, $\mathbf{x}_k = \varphi_k(\mathbf{X})$ and $\mathbf{x}_{k+1}$ are the current nodal coordinates at iterations *k* and *k+1*, respectively. Ideally, Newton's method achieves a quadratic convergence rate in the neighborhood of a solution. However, it also requires the formation and factorization of the stiffness matrix at every iteration, which is the most costly part of nonlinear finite element analysis from a computational standpoint. Quasi-Newton methods offer an attractive alternative. For these methods, the true stiffness matrix **K** is replaced by an approximation that is relatively easy to calculate. FEBio uses the BFGS method [23], which is one of the more effective quasi-Newton methods for computational solid mechanics.

2.4.2 Element technology

The 3D solid elements in FEBio are isoparametric elements, meaning that the element geometry and displacement fields on the element are interpolated with the same shape functions. Currently available elements include the linear hexahedron, pentahedron, and tetrahedron. It is well-known that these elements tend to "lock" for nearly and fully incompressible material response when using a displacement-only formulation. FEBio deals with this problem by using a three-field element formulation (the mean dilatation method) for the hexahedral and pentahedral elements. [25]. For tetrahedral elements, a nodally integrated tetrahedron provides enhanced performance for finite deformation and near-incompressibility compared to the standard constant strain tetrahedron [26].

Quadrilateral and triangular shell elements in FEBio use an extensible director formulation [27]. Six degrees of freedom are assigned to each shell node: three displacement degrees of freedom and three director degrees of freedom. A three-point Gaussian quadrature rule is used for integration through the thickness of the shell.

FEBio also solves porous media problems using the mixture framework known as the biphasic theory [28]. For such problems, a fluid phase exists along with the solid phase and therefore requires the solution of a separate fluid field problem. In FEBio, a $\mathbf{u}-p$ formulation is used to solve the coupled solid-fluid problem for the unknown displacements and fluid pressures [29]. Like the displacements, the fluid pressures are defined at the nodes and the biphasic elements are integrated using the same quadrature rule as the structural elements. In the formulation, an additional equation is needed to satisfy the conservation of mass for the mixture. The weak formulation now requires the solution of a coupled displacement-pressure problem.

$$\delta W_m\left(\phi, p, \delta\mathbf{v}\right) = \int_v \boldsymbol{\sigma} : \delta\mathbf{d}dv - \int_v \mathbf{f} \cdot \delta\mathbf{v}dv - \int_{\partial v} \mathbf{t} \cdot \delta\mathbf{v}da = 0, \tag{2.6}$$

and

$$\delta W_f\left(\phi, p, \delta p\right) = \int_v \left[\mathbf{w} \cdot \nabla\left(\delta p\right) - \delta p\mathbf{1} : \mathbf{d}\right]dv - \int_{\partial v} \delta p\mathbf{w} \cdot \mathbf{n}\, da = 0. \tag{2.7}$$

Equation (2.6) is recognized as the virtual work Equation (2.1). In Eq. (2.7), δp is a virtual fluid pressure and $\mathbf{w}$ is the flux of fluid relative to the solid, which can be calculated for instance by Darcy’s law $\mathbf{w} = -\mathbf{k}\nabla p$ where $\mathbf{k}$ is the permeability tensor. $\mathbf{n}$ is the unit outward normal to ∂v.

2.4.3 Constitutive models

The material representations in FEBio span the range of elastic solids, viscoelastic solids, biphasic materials, biphasic-solid materials, and rigid bodies. Many of the constitutive models are based on the concept of hyperelasticity. Constitutive models with isotropic, transversely isotropic [30], and orthotropic material symmetries are available.

For most materials, uncoupled formulations are used so that these materials can be used effectively in nearly-incompressible analyses. Detailed descriptions of all of the materials in FEBio can be found in the FEBio Theory Manual [21].

2.4.4 Contact

Several contact algorithms are available in FEBio. To enforce the contact constraint, a contact work contribution is added to the virtual work statement in Eq. (2.1). The contact integral is of the following form:

$$G^c = -\int_{\Gamma_c^{(1)}} \mathbf{t}^{(1)}(\mathbf{x}) \cdot \left[w^{(1)}(\mathbf{x}) - w^{(2)}\left(\bar{\mathbf{y}}(\mathbf{x})\right) \right] d\Gamma \,. \tag{2.8}$$

Here, $\mathbf{t}^{(i)}$ is the contact reaction force on body i and $w^{(i)}$ are weight functions. One can also look at $\mathbf{t}$ as a Lagrange multiplier that enforces the contact constraint. This contact formulation is general enough to support both sliding interfaces, where the objects may slide across each other, and tied interfaces, where both objects are required to stick together at the contact interface. In the case of porous permeable media, a new contact algorithm between biphasic materials allows for tracking of fluid flow across the contact interface. In other words, fluid can flow from one side of the contact interface to the other when both contact surfaces are biphasic [31]. FEBio supports both a standard penalty-type enforcement of the contact constraint as well as the augmented Lagrange method, which calculates the Lagrange multipliers incrementally [32].

2.4.5 Rigid bodies

Rigid bodies are supported as a separate material type. For groups of elements, faces, and/or nodes that are assigned the “rigid body” material type in FEBio, nodal degrees of freedom are condensed down to six degrees of freedom. The result is a vast reduction in the number of the total degrees of freedom and leads to a very efficient implementation of the rigid body constraint [33]. Rigid bodies can also be connected by rigid joints. The rigid joint constraints are enforced using an augmented Lagrangian method.

2.4.6 Linear solvers

Since over 90% of the execution time of large problems in an implicitly integrated nonlinear FE code is spent in the linear solver (which is called repeatedly for the Newton or quasi-Newton solution method), the linear solver is typically the bottleneck for large computations. Consequently, an efficient and robust linear solver was needed. The linear solver must run on all supported platforms (Linux, Windows, Mac), take optimal advantage of the sparsity of the stiffness matrix, handle both symmetric and unsymmetric matrices (necessary for biphasic problems), and support parallel execution. For this reason, FEBio includes support for Pardiso [34,35], SuperLU [36], and WSMP [37]. These solvers take optimal advantage of the sparse stiffness matrix by storing only the nonzero matrix elements using the Harwell-Boeing matrix storage format. Further, they can handle nonsymmetric matrices and support parallel execution on multicore shared-memory architectures.

2.4.7 Software implementation

FEBio is written in C++. This programming language was chosen over FORTRAN – the more commonly used language in scientific computing – for two reasons. First, the performance penalties that once existed in object-oriented programming languages (compared to procedural languages such as FORTRAN) have been mostly overcome by modern compilers [38]. Second, C++ encourages the use of a modular code structure that simplifies the development and maintenance of a sophisticated software program. This modular structure also offers advantages for users who wish to customize FEBio for their own needs. For instance, it provides a simple yet elegant way to add new materials without the need to modify any of the existing code. By simply providing a new constitutive relation, the new material is automatically recognized in the input format and supported in restart analyses, the parameter optimization module, as well as all other features of FEBio that support different materials. To simplify the addition of custom materials further, a custom math library was designed that facilitates the implementation of tensor and matrix operations. This library allows the user to implement the material model using programming statements that closely resemble the analytical mathematical expressions. To verify the material implementation, several diagnostic tools are available for the developer to check the consistency between the stress tensor and the fourth-order elasticity tensor. The software is commented extensively to clarify the data flow of the code. Online documentation of the source code is available as well.

2.5 Verification Problems

Results produced by FEBio were compared to analytical solutions or the solutions from two established and verified FE codes, ABAQUS (Version 6.7, Simulia, Providence, RI) and NIKE3D (Version 3.4.1, Methods Development Group, Lawrence Livermore National Laboratory, Livermore, CA [39]). The problem description and results are presented together for each verification problem. The FEBio Verification Suite consists of over 140 test problems, and those described below are representative.

2.5.1 Uniaxial tension-compression material tests

The single-element uniaxial stress test is a standard verification test for material models. Since the deformation is homogeneous, the analytical solution can often be obtained in closed form. This allows for direct comparison of the FEBio results with the analytical solution. In this case study, results for the Mooney-Rivlin and Ogden hyperelastic constitutive models are presented. Both materials are often used to represent the matrix component of biological tissues. The Mooney-Rivlin constitutive model in FEBio has the following uncoupled strain-energy function:

$$W^{MR} = C_1\left(\tilde{I}_1 - 3\right) + C_2\left(\tilde{I}_2 - 3\right) + \frac{K}{2}\left(\ln J\right)^2, \tag{2.9}$$

where C_1 and C_2 are material parameters, K is the bulk modulus, $\tilde{I}_1$ and $\tilde{I}_2$ are the 1st and second deviatoric invariants of the right Cauchy-Green deformation tensor $\mathbf{C} = \mathbf{F}^T\mathbf{F}$, $\mathbf{F}$ is the deformation gradient, and $J = \det(\mathbf{F})$ is the Jacobian of the deformation. The material parameters were chosen to be $C_1 = 6.8$ MPa, $C_2 = 0$, which approximates the material parameters of articular cartilage during fast loading [40]. The value of K was chosen sufficiently high to satisfy near-incompressibility ($K \sim 10^5$ MPa).

The Ogden constitutive model with uncoupled deviatoric and dilatational response has the following strain-energy function [41]:

$$W^{OG}\left(\lambda_1, \lambda_2, \lambda_3, J\right) = \sum_{i=1}^{N} \frac{c_i}{m_i^2}\left(\tilde{\lambda}_1^{m_i} + \tilde{\lambda}_2^{m_i} + \tilde{\lambda}_3^{m_i} - 3\right) + \frac{K}{2}\left(\ln J\right)^2. \tag{2.10}$$

Here, $\tilde{\lambda}_i$ are the deviatoric principal stretches and c_i and m_i are material coefficients.

Material incompressibility was enforced using the augmented Lagrangian method. Material coefficients were chosen to represent the bulk soft tissue properties of the heelpad (N = 1, c_1 = 0.0329 MPa, m_1 = 6.82) [42]. Again, the bulk modulus K was chosen sufficiently high to enforce near-incompressibility. Results from FEBio were compared to analytical solutions. Excellent agreement was achieved between the analytical solutions and the predicted responses from FEBio for both constitutive equations (Fig. 2.1).

2.5.2 Confined compression creep

The confined compression creep test is often used for material characterization of deformable porous media [28,43,44]. A cylindrical tissue sample is placed in a confining chamber with rigid impermeable bottom and side walls. The top of the tissue is compressed with a free-draining rigid porous indenter. If a constant applied load is used, a creep deformation is produced, increasing over time (Fig. 2.2).

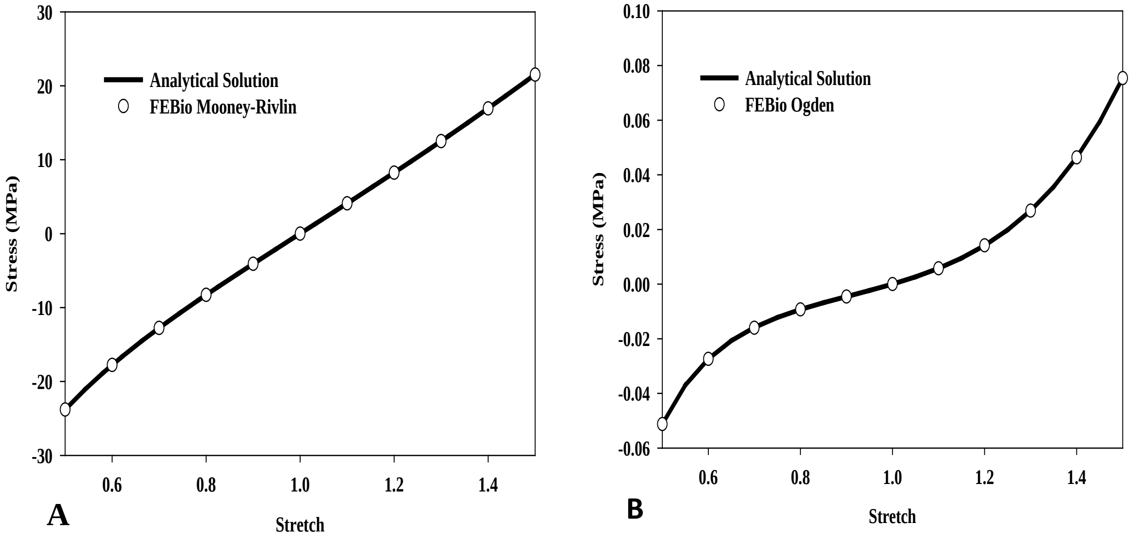


Fig. 2.1: Simulation of uniaxial tension-compression material tests for the Mooney Rivlin (top) and Ogden (bottom) constitutive models. There was excellent agreement between the analytical solutions and the predictions from FEBio for both constitutive equations.

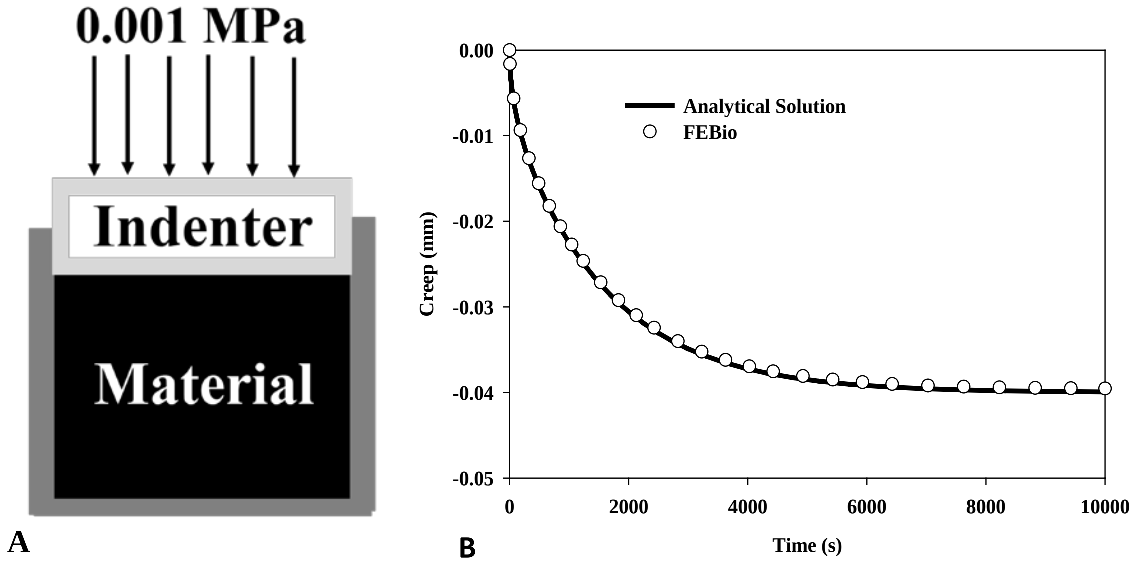


Fig. 2.2: Confined compression creep testing of deformable porous media. Top - loading and boundary conditions for the confined compression creep simulation. Bottom – creep displacement as a function of time, calculated from the analytical solution and predicted by FEBio, showing the excellent agreement between the two results.

A unit cube was created and the nodes were constrained so that only the top surface could move along the normal direction. A pressure of 0.001 MPa was applied to the top surface. The material was assumed to be biphasic. The solid phase was modeled as a neo-Hookean material with strain energy given by:

$$W^{NH} = \frac{\mu}{2}\left(I_1 - 3\right) - \mu \ln J + \frac{\lambda}{2}\left(\ln J\right)^2 . \qquad (2.11)$$

Here, μ and λ are the Lamé parameters, $I_1 = \mathrm{tr}\,\mathbf{C}$ is the first invariant of the right Cauchy-Green deformation tensor. For this problem, the material parameters were chosen to be $\lambda = 1.43$MPa, $\mu = 0.357$ MPa and the permeability of the mixture was chosen to be 10^{-3}mm^4/N.s. Results from FEBio were compared to the analytical solution in Mow et al. [28] for verification. The creep response predicted by FEBio was in excellent agreement with the analytical solution for the confined compression problem (Fig. 2.2).

2.5.3 Unconfined compression stress relaxation

In this test, a cylindrical tissue sample is exposed to a prescribed displacement in the axial direction while left free to expand radially. After loading the tissue, the displacement is held constant while the tissue under the displacement relaxes in the radial direction due to interstitial fluid flow through the material. For porous media, the outer radial boundary is free-draining (Fig. 2.3, arrows).

A quarter-symmetry mesh was used to model the cylindrical tissue sample. The outer radius and height of the cylinder were 1 mm, while the axial compression was 0.01 mm. The bottom of the tissue was constrained and quarter-symmetry boundary conditions were applied. The fluid pressure was constrained to zero at the outer radial surface, while the fluid pressure throughout the rest of the body was determined by solution of the

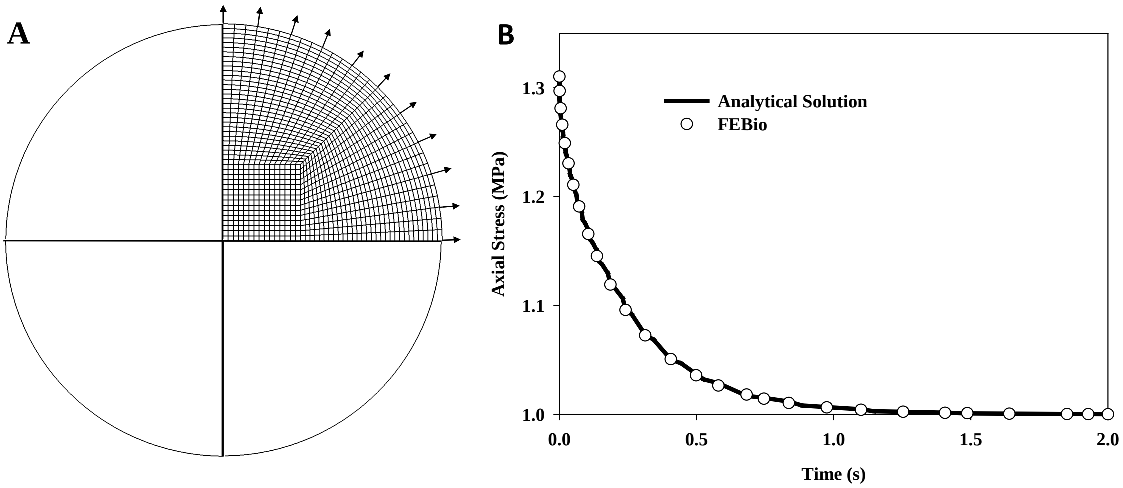


Fig. 2.3: Unconfined compression stress relaxation. Top - quarter-symmetry model used for the simulation with the free fluid flow through the outer boundaries indicated with arrows. Bottom - axial stress versus time calculated from the analytical solution and predicted by FEBio, showing the excellent agreement between the two results.

equilibrium equations for porous media. The solid phase was represented as a neo-Hookean material (Eq. (2.11), $\lambda = 0.186\,\text{MPa}$, $\mu = 0.435\,\text{MPa}$) and the permeability of the mixture was chosen to be $10^{-3}\,\text{mm}^4/\text{N}\cdot\text{s}$.

The unconfined compression response can be modeled with the biphasic theory [28]. For the special case of a cylindrical geometry and assumptions regarding the direction of the fluid flow, Armstrong et al. [45] found a closed-form analytical solution for the average axial stress on the sample in response to a step loading function. The axial stress predicted by FEBio was nearly identical to the analytical solution provided by Armstrong et al. (Fig. 2.3).

2.5.4 Strip biaxial stretching of an elastic sheet with a circular hole

This example demonstrates the use and verifies the results of a hyperelastic material in plane stress under large deformation. A thin, initially square sheet containing a centrally located circular hole was subjected to strip biaxial stretch. The undeformed square sheet was 165×165×2 mm, with a centrally located internal hole of radius 6.35 mm (Fig. 2.4). The sheet was meshed with 32 hexahedral elements. This example is identical to the problem in section 1.1.8 of the ABAQUS Benchmark Manual (Version 6.7).

The sheet was stretched to a length of 1181 mm (615.76% strain), while the edges parallel to the displacement were restrained from contracting. The sheet was represented as a hyperelastic Mooney-Rivlin material with uncoupled deviatoric and volumetric behavior (Eq (2.9)). The material coefficients were $C_1 = 0.1863$ MPa, $C_2 = 0.00979$ MPa, and K = 100 MPa, as used by Oden [46] to match the experimental results for a rubber sheet first reported by Treloar [47]. The reaction force predicted by FEBio was compared to results from ABAQUS and NIKE3D as a function of percent elongation.

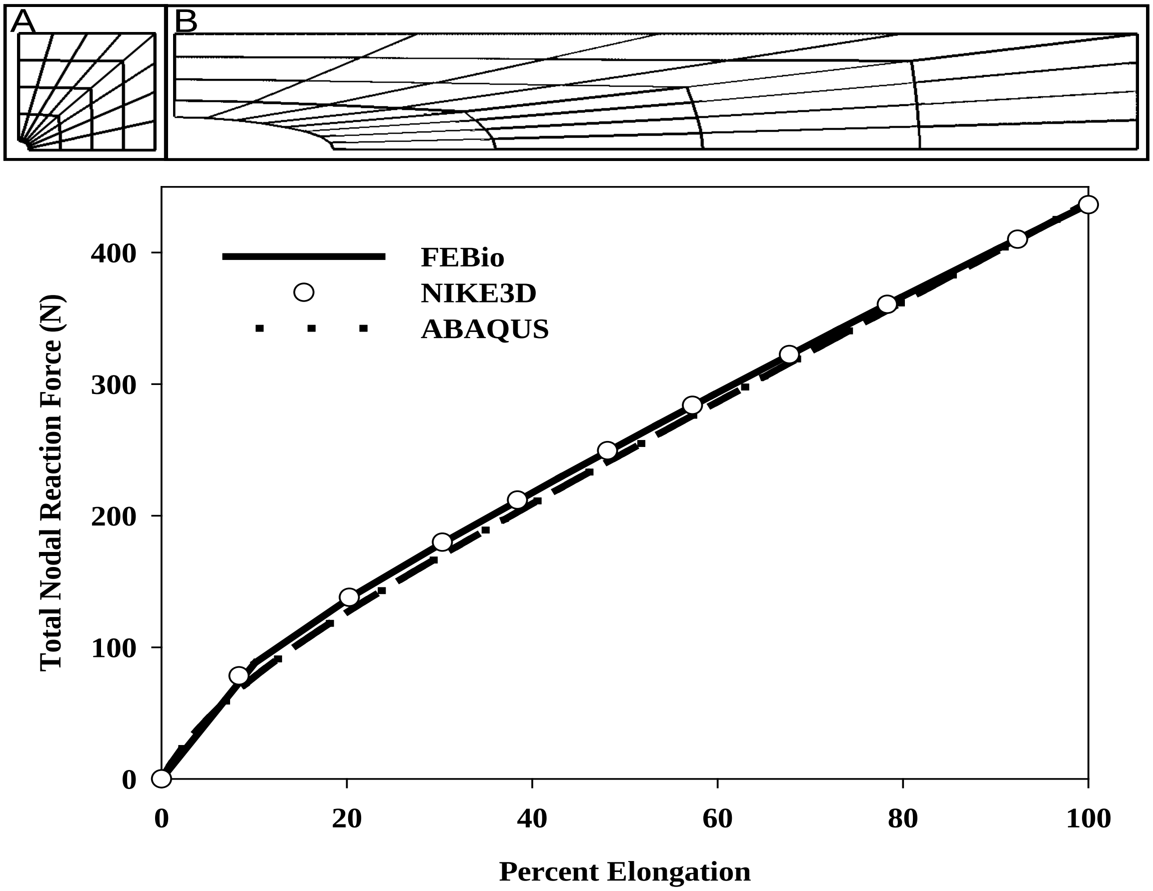


Fig. 2.4: Strip biaxial stretching of an elastic sheet with a circular hole. Top – Panel A, quarter-symmetry model used for this simulation. Panel B, deformed configuration after applied strain. Bottom - total nodal reaction force versus percent elongation for FEBio, NIKE3D, and ABAQUS. FEBio and NIKE3D predicted identical results that were slightly different than ABAQUS due to the different algorithms used to enforce the material incompressibility.

The total nodal reaction force predicted by FEBio was identical to the results produced by NIKE3D, but was slightly higher than predictions from ABAQUS (Fig. 2.4). The small difference is likely due to the slightly different implementation of the Mooney-Rivlin material in ABAQUS. More specifically, ABAQUS uses a different function for the dilatational strain energy (that is, the last term in Eq. (2.9)) than FEBio and NIKE3D.

2.5.5 Geometrically nonlinear analysis of a cantilever beam

A 10 m long cantilever beam was subjected to a tip load in the transverse direction to produce over 8 m of lateral deflection (Fig. 2.5). The beam had a solid rectangular cross-section (100 mm thick × 150 mm height) and three different mesh densities were used (100, 200, and 400 elements along the length of the beam). One element was used through the thickness in all cases. One end of the beam was constrained using nodal constraints, while a vertical load of 269.35 N was applied at the opposite tip of the beam. The beam was represented as a hyperelastic St. Venant-Kirchhoff elastic material:

$$W^{SV} = \frac{1}{2}\lambda\left(\mathrm{tr}\,\mathbf{E}\right)^2 + \mu\left(\mathbf{E}:\mathbf{E}\right), \tag{2.12}$$

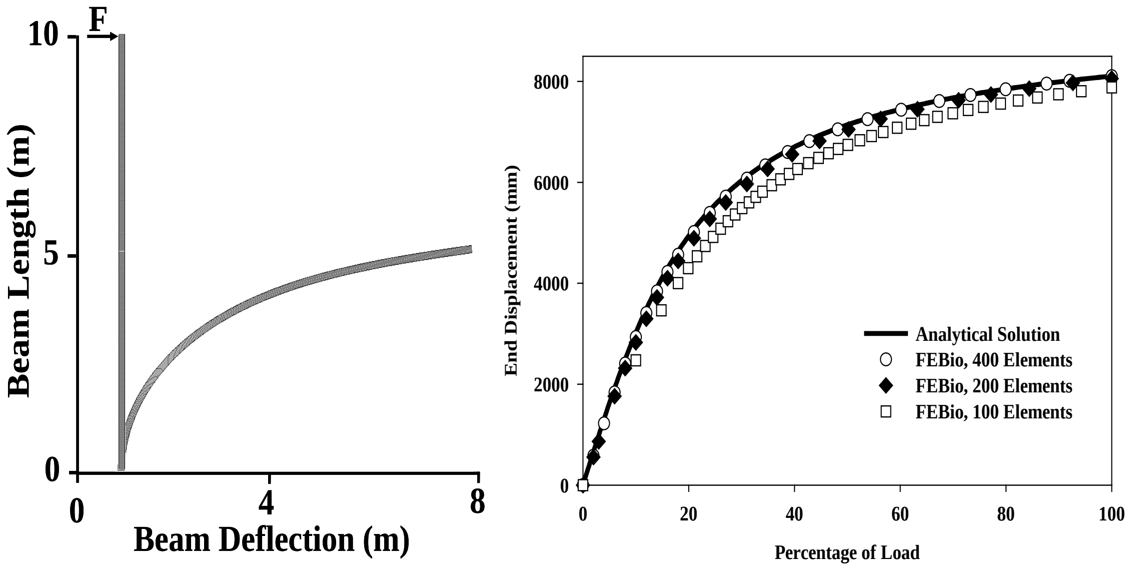


Fig. 2.5: Geometrically nonlinear analysis of a cantilever beam. Top – cantilever beam in the undeformed and deformed configurations showing the beam length and deflection. Bottom – end displacement versus percentage of applied load for the analytical solution and FEBio simulations using 100, 200, and 400 elements along the length of the beam. The beam deflection predicted by FEBio was nearly identical to the analytical solution when 400 elements were used to discretize the beam.

where λ and μ are the Lamé parameters ($\lambda = 0$, $\mu = 50$ MPa) and **E** is the Green-Lagrange strain tensor. The use of the Green-Lagrange strain tensor in the formulation of the St. Venant-Kirchhoff constitutive equation makes it appropriate for the large rotations and small strains in this problem. The cantilever tip deflection predicted by FEBio was compared to the analytical solution for the large deflection of a cantilever beam [48].

The beam deflection predicted by FEBio was nearly identical to the analytical solution for the geometrically nonlinear analysis of a cantilever beam for a model with 400 elements along the length of the beam (Fig. 2.5). Models with 100 and 200 elements produced a slightly stiffer response.

2.5.6 Twisted ribbon test for shells

This is one of a number of verification problems that are used to verify the shell element formulation in FEBio. A plate discretized with shell elements was twisted by constraining one end and applying equal but opposite forces to the corners of the non-constrained end (Fig. 2.6). The problem description and analytical solution were originally presented by Batoz [49]. The plate dimensions were 1.00×2.00×0.05 mm and five models with mesh densities from 32 to 2048 elements were used for a convergence study. Nodes along one short edge were clamped using nodal constraints and the plate was twisted by applying equal but opposite forces of 1 N to the corners nodes of the opposite short edge. The St. Venant-Kirchhoff material was used (Eq. (2.12), $\lambda = 2.5\times10^7$ MPa, $\mu = 2\times10^6$ MPa). The displacement of the corner node predicted by FEBio was compared to the analytical solution.The corner displacement predicted by FEBio was nearly identical to the analytical solution for the twisted ribbon test for shells when 2048 elements were used to mesh the plate (Fig. 2.6). To put this result in perspective, the corner displacement predicted by

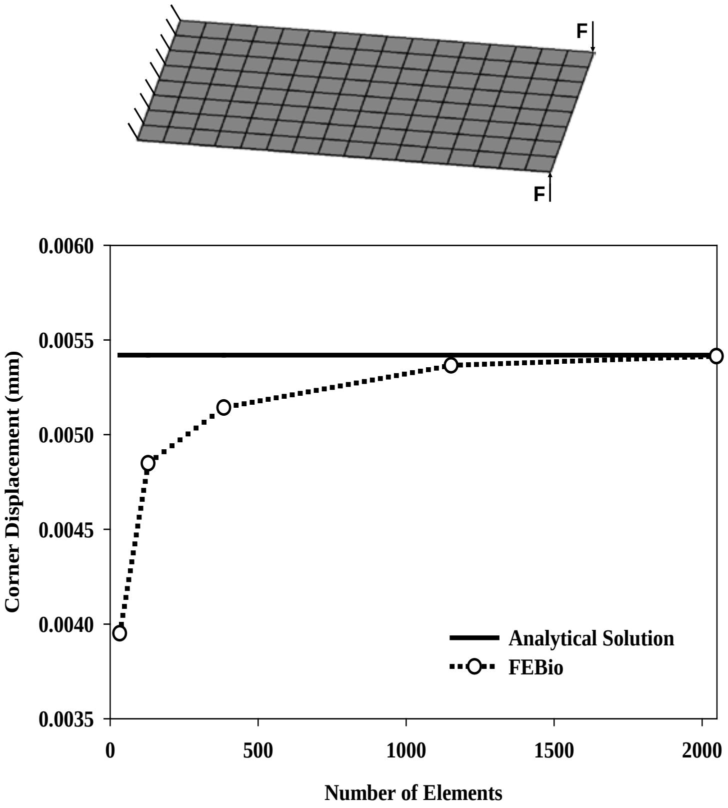


Fig. 2.6: Twisted ribbon test for shells. Top - model in the undeformed configuration with loading and boundary conditions indicated. Bottom - displacement of the outside corner node versus number of elements used to discretize the model. 2048 elements were needed to exactly match the analytical solution, but there was <2% difference when only half that many elements were used to discretize the model.

FEBio was <2% different when half that many elements were used and ~30% different when only 32 elements were used.

2.5.7 Upsetting of an elastic billet

This problem is commonly used to test finite element approaches for enforcing near- and full-incompressibility, since the domain is highly constrained. A billet of material was compressed between two rigid flat surfaces (Fig. 2.7A). Since the material was only compressed to the point when the material bulging from the middle of the billet makes contact with the flat surfaces (Fig. 2.7B), this problem is a simplification of the contact

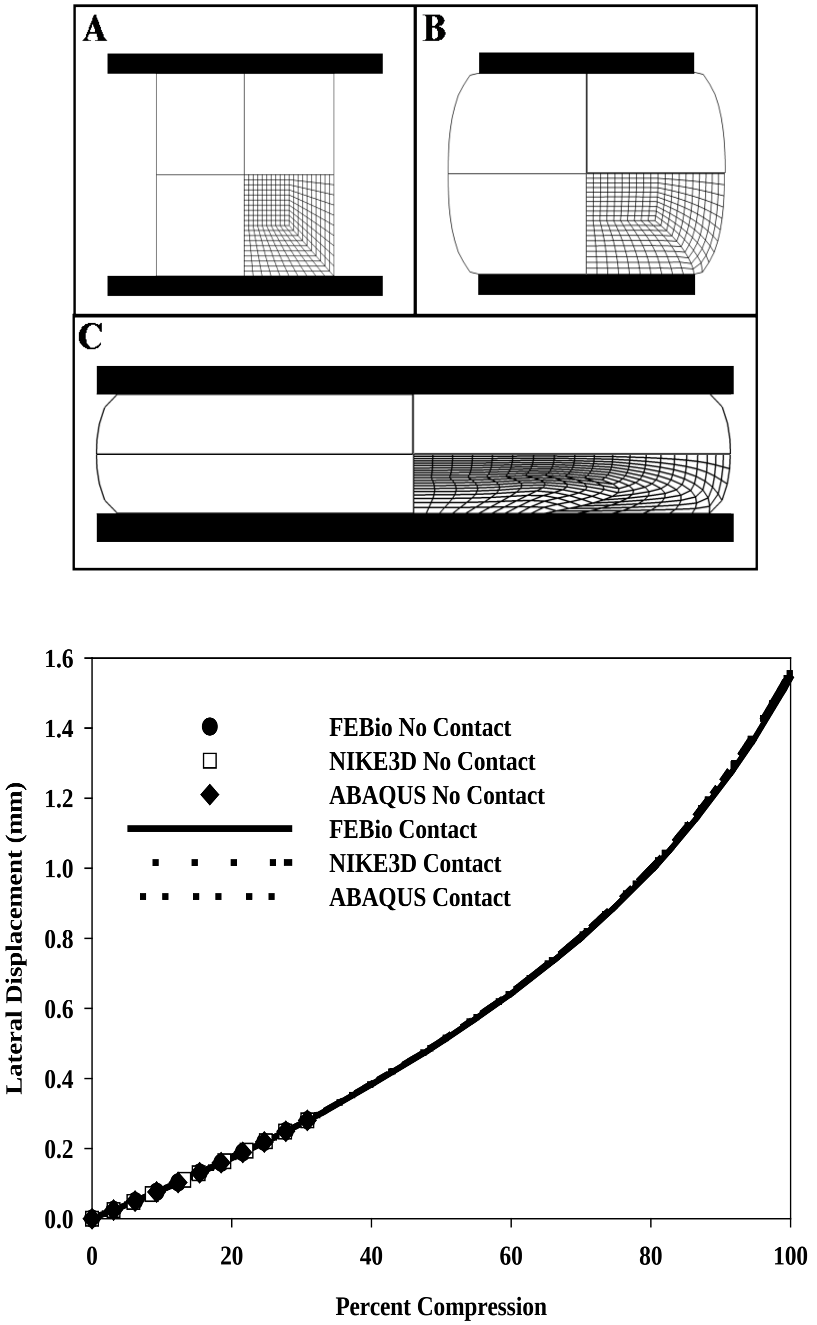


Fig. 2.7: Upsetting of an elastic billet. Top – Panel A, quarter-symmetry model in the initial configuration. Panel B, billet deformed until there is contact between the inner material of the billet and the rigid planes. Panel C, deformed billet with contact between the inner material and the rigid planes. Bottom – lateral displacement (bulge) versus percent compression for FEBio, NIKE3D, and ABAQUS. There was excellent agreement in the predicted results from all three FE codes for both the contact and noncontact versions of this problem.

problem presented in the next example (Fig. 2.7C).

A quarter-symmetry, plane strain model was produced (Fig. 2.7A). The model dimensions were 1.0×1.0×0.1 mm, and a mesh of 10×10×1 elements produced converged results. Boundary and loading conditions were prescribed using nodal constraints and nodal loads, respectively. The billet was represented as an uncoupled Mooney-Rivlin material (Eq. (2.9), C_1=1 MPa, C_2=10 MPa, K=10,000 MPa). The augmented Lagrangian method was used to enforce incompressibility [32]. The displacement of the top right node of the mesh (representing the amount of lateral bulging) was plotted versus the percent compression and the results from FEBio were compared to results from ABAQUS and NIKE3D. The lateral displacement of the billet predicted by FEBio was nearly identical to the results produced by ABAQUS and NIKE3D (Fig. 2.7, right panel).

To verify the contact algorithm implemented in FEBio, contact between the incompressible material of the billet and the rigid plane was added to the problem. The billet was compressed 60% of its original height, causing extensive contact between the rigid plane and the material originally in the middle of the billet. The boundary conditions for this problem were the same as the previous problem, but here the compression of the billet was caused by frictionless contact between the billet and rigid plane. This contact was enforced using an augmented-Lagrangian method. Again, the lateral displacement of the billet predicted by FEBio was nearly identical to the results produced by ABAQUS and NIKE3D. The results from the noncontact and contact problems lined up through the smaller range of compression (Fig. 2.7, right panel).

2.5.8 Crushing of a pipe

This problem simulated the crushing of a long, straight pipe of cylindrical cross-section between two flat, frictionless anvils. This benchmark problem includes large strains/rotations and contact between deformable and rigid materials. A quarter-symmetry, plane strain model of a 114.3 mm radius pipe with an 8.87 mm thickness was created for this problem (Fig. 2.8). The model was meshed with 24 elements along the arc, four elements through the thickness, and one element through the length. Nodal constraints were used for the plane-strain and quarter-symmetry boundary conditions, and the problem was driven by prescribing a 50 mm displacement to the rigid body. Contact between the anvil and pipe was enforced using the augmented Lagrangian method. The St. Venant-Kirchhoff material was used for this problem ($\lambda = 107$ GPa, $\mu = 71.5$ GPa). The rigid body reaction force predicted by FEBio was compared to the results from NIKE3D and ABAQUS as a

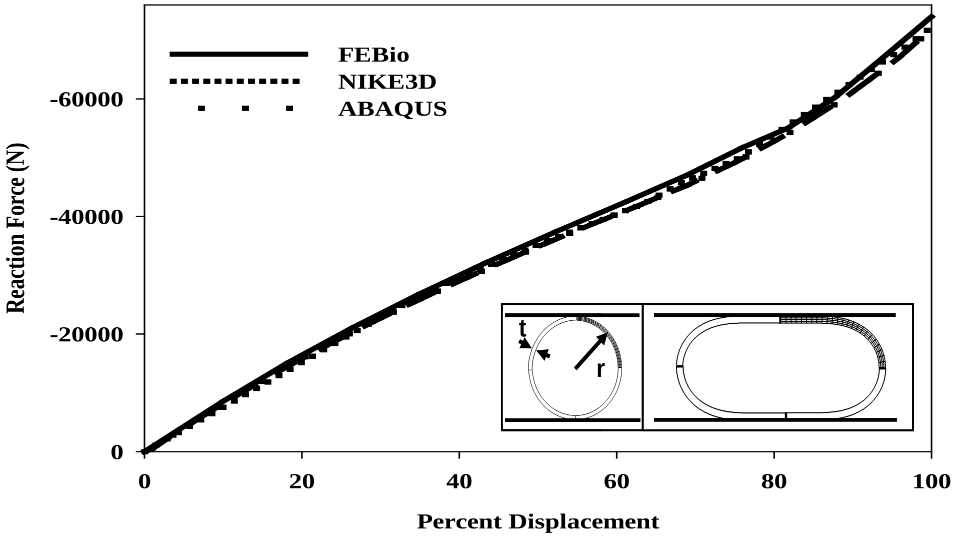


Fig. 2.8: Crushing of a pipe. Top, Left - quarter-symmetry model with rigid crushing planes and key dimensions shown. Top, Right - pipe in the final crushed configuration. Bottom - rigid body reaction force versus percent applied displacement for the crushing planes. There was less than 3% difference in the peak forces predicted by the three FE codes.

function of anvil displacement.

The results predicted by the three FE codes are shown in (Fig. 2.8), demonstrating good agreement among all three codes. Small observed differences were most likely due to the different algorithms to enforce contact. Still, at peak displacement there was less than 3% difference in the predicted forces.

2.5.9 Cartilage layer compressed by a flat, rigid, impermeable surface

A half-symmetry finite element frictionless contact analysis was performed between a spherical elastic layer anchored to a rigid impermeable substrate and a flat impermeable rigid surface (Fig. 2.9). The geometry was representative of the articular layer of an immature bovine humeral head, with a cartilage surface radius of 46.3 mm and a cartilage layer thickness of 0.8 mm.The converged mesh consisted of 20 tri-linear hexahedral elements through the thickness, 50 along the radial direction, and 14 along the circumferential direction. The top of the cartilage layer was constrained by defining the top layer of nodes as a rigid body with all 6 DOF constrained. The deformation at the center of the articular layer was set to 0.095 mm (~12% of the thickness) by displacing the bottom nodes using nodal prescribed displacements. The cartilage layer was represented by the Mooney-Rivlin constitutive model in Eq. (2.9) to approximate the short-time response of the tissue, again using $C_1 = 6.8$ MPa and $C_2 = 0$,. The radial and circumferential stresses predicted by FEBio through the thickness of the middle of the cartilage layer were compared to the results from ABAQUS and NIKE3D.The predicted radial and circumferential stresses through the thickness of the middle of the cartilage layer were the

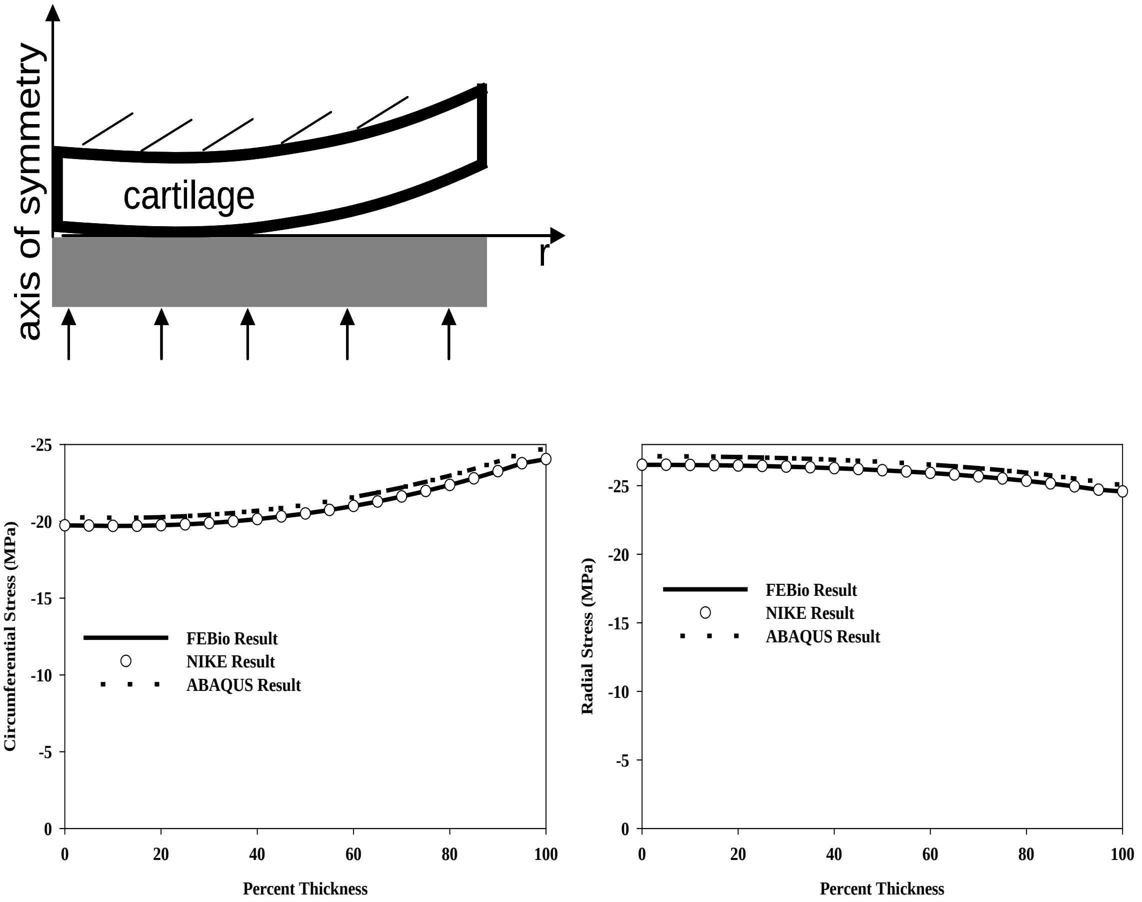


Fig. 2.9: Cartilage layer compressed by a flat, rigid, impermeable surface. Top - schematic of the model illustrating the boundary and loading conditions. Middle - radial stress versus percent thickness through the middle of the cartilage layer predicted by FEBio, NIKE3D, and ABAQUS. Bottom - circumferential stress versus percent thickness predicted by the three FE codes. The stresses predicted by FEBio and NIKE3D were nearly identical and the stresses predicted by ABAQUS were less than 3% different than the other codes.

same for FEBio and NIKE3D, but the stresses predicted by ABAQUS were consistently slightly larger (Fig. 2.9).The small differences (<3%) are most likely due to differences in the algorithms used to enforce the contact constraint: FEBio and NIKE3D use an augmented Lagrangian method to enforce contact while ABAQUS uses a Lagrange multiplier method.

2.5.10 Validated model of the human inferior glenohumeral ligament

This problem illustrates the use of FEBio for a current research topic in computational biomechanics. The objective of the research was to develop and validate a model of the inferior glenohumeral ligament using experimental data. The subject-specific model incorporated experimental measurements of the geometry of capsular regions, humerus, and scapula, the mechanical properties of capsular regions; and joint kinematics during a simulated clinical exam. Strain distributions were measured experimentally and used to validate the FE model predictions. A full description of the model and its validation were reported by Moore et al. [50]. This original publication made use of NIKE3D for the finite element analysis. At the time, only hypoelastic materials could be used with shell elements in NIKE3D to represent the capsule. We re-analyzed this model in FEBio, which has allowed the use of a variety of more appropriate and accurate constitutive models for the capsule material (Fig. 2.10) [51].

2.6 Discussion

This paper presented FEBio, a new software tool for computational biomechanics. Some of the underlying theoretical aspects were discussed. In particular, the two most important areas of application were mentioned, namely large deformation and contact of

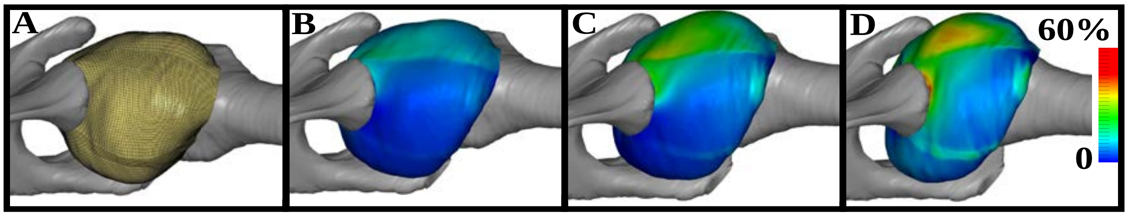


Fig. 2.10: Validated model of the human inferior glenohumeral ligament, analyzed with FEBio. A) Initial position of the model, showing the fine discretization of the shoulder capsule with shell elements. Panels B-D), first principal Green-Lagrange

solid and biphasic materials. FEBio provides analysis methods and constitutive models that are relevant for computational biomechanics and in this regard differs from many of the existing software tools that are currently in use for research. Whereas users must tailor their use of commercial applications around the limitations of the software, FEBio offers them a way to tailor the software to their specific needs.

A set of test problems was presented to illustrate our approach to verification of the presented algorithms. The results demonstrated very good to excellent agreement with either analytical solutions and/or results obtained from other finite element software. Small discrepancies were easily explained by differences in numerical algorithms between the different software packages. These results give researchers confidence that FEBio can provide accurate results for their computational research questions as well. Of course, problem-specific verification, mesh convergence studies and validation are still necessary to assure that the predictions from FEBio (or any other FE code) are accurate for a specific problem [16,17].

The development of FEBio is ongoing. New element formulations, contact algorithms, and constitutive equations are being implemented and will become available in future versions. Continued emphasis will be placed on support and dissemination of FEBio in the form of documentation of the software and its example problems, online manuals, and on our online forum. We hope that this open-source finite element framework using modern software design principles will not only provide a new and useful tool for computational biomechanics, but also set a new standard for computational simulation software in this field.

2.7 Acknowledgments

Financial support from the National Institutes of Health (1R01GM083925) is gratefully acknowledged. The authors thank Drs. Ahmet Erdemir (Cleveland Clinic Foundation) and Alexander Veress (University of Washington) for their feedback related to FEBio software development.

2.8 References

[1] Anderson, A. E., Peters, C. L., Tuttle, B. D., and Weiss, J. A., 2005, “SubjectSpecific Finite Element Model of the Pelvis: Development, Validation and Sensitivity Studies,” J. Biomech. Eng., 127, pp. 364–373.

[2] Spilker, R. L., Feinstein, J. A., Parker, D. W., Reddy, V. M., and Taylor, C. A., 2007, “Morphometry-Based Impedance Boundary Conditions for PatientSpecific Modeling of Blood Flow in Pulmonary Arteries,” Ann. Biomed. Eng., 35, pp. 546–559.

[3] Wang, L., Zhang, H., Shi, P., and Liu, H., 2006, “Imaging of 3D Cardiac Electrical Activity: A Model-Based Recovery Framework,” Med. Image Comput. Comput. Assist. Interv., 9, pp. 792–799.

[4] Speelman, L., Bohra, A., Bosboom, E. M., Schurink, G. W., van de Vosse, F. N., Makaorun, M. S., and Vorp, D. A., 2007, “Effects of Wall Calcifications in Patient-Specific Wall Stress Analyses of Abdominal Aortic Aneurysms,” J. Biomech. Eng., 129, pp. 105–109.

[5] Portnoy, S., Yarnitzky, G., Yizhar, Z., Kristal, A., Oppenheim, U., Siev-Ner, I., and Gefen, A., 2007, “Real-Time Patient-Specific Finite Element Analysis of Internal Stresses in the Soft Tissues of a Residual Limb: A New Tool for Prosthetic Fitting,” Ann. Biomed. Eng., 35, pp. 120–135.

[6] Schmid-Schonbein, G. W., and Diller, K. R., 2005, “Transport Processes in Biomedical Systems: A Roadmap for Future Research Directions,” Ann. Biomed. Eng., 33, pp. 1136–1141.

[7] Crawford, R. P., Rosenberg, W. S., and Keaveny, T. M., 2003, “Quantitative Computed Tomography-Based Finite Element Models of the Human Lumbar Vertebral Body: Effect of Element Size on Stiffness, Damage, and Fracture Strength Predictions,” J. Biomech. Eng., 125, pp. 434–438.

[8] Li, L. P., and Herzog, W., 2006, “Arthroscopic Evaluation of Cartilage Degeneration Using Indentation Testing—Influence of Indenter Geometry,” Clin. Biomech. (Bristol, Avon), 21, pp. 420–426.

[9] Li, L. P., and Herzog, W., 2005, “Electromechanical Response of Articular Cartilage in Indentation—Considerations on the Determination of Cartilage Properties During Arthroscopy,” Comput. Methods Biomech. Biomed. Eng., 8, pp. 83–91.

[10] Davids, N., and Mani, M. K., 1974, “A Finite Element Analysis of Endothelial Shear Stress for Pulsatile Blood Flow,” Biorheology, 11, pp. 137–147. Available at http://www.ncbi.nlm.nih.gov/pubmed/4441640.

[11] Doyle, J. M., and Dobrin, P. B., 1971, “Finite Deformation Analysis of the Relaxed and Contracted Dog Carotid Artery,” Microvasc. Res., 3, pp. 400–415.

[12] Janz, R. F., and Grimm, A. F., 1972, “Finite-Element Model for the Mechanical Behavior of the Left Ventricle. Prediction of Deformation in the PotassiumArrested Rat Heart,” Circ. Res., 30, pp. 244–252.

[13] Matthews, F. L., and West, J. B., 1972, “Finite Element Displacement Analysis of a Lung,” J. Biomech., 5, pp. 591–600.

[14] Farah, J. W., Craig, R. G., and Sikarskie, D. L., 1973, “Photoelastic and Finite Element Stress Analysis of a Restored Axisymmetric First Molar,” J. Biomech., 6, pp. 511–520.

[15] Belytschko, T., Kulak, R. F., Schultz, A. B., and Galante, J. O., 1974, “Finite Element Stress Analysis of an Intervertebral Disc,” J. Biomech., 7, pp. 277–285.

[16] Anderson, A. E., Ellis, B. J., and Weiss, J. A., 2007, “Verification, Validation and Sensitivity Studies in Computational Biomechanics,” Comput. Methods Biomech. Biomed. Eng., 10, pp. 171–184.

[17] Henninger, H. B., Reese, S. P., Anderson, A. E., and Weiss, J. A., 2010, “Validation of Computational Models in Biomechanics,” Proc. Inst. Mech. Eng. Part H, J. Eng. Med., 224, pp. 801–812.

[18] Maas, S. A., and Weiss, J. A., 2007, FEBio User’s Manual, http://mrl.sci.utah. edu/software/febio.

[19] ASME, 2006, “Guide for Verification and Validation in Computational Solid Mechanics,” ASME Committee (PT60) on Verification and Validation in Computational Solid Mechanics.

[20] Babuska, I., and Oden, J. T., 2004, “Verification and Validation in Computational Engineering and Science: Basic Concepts,” Comput. Methods Appl. Mech. Eng., 193, pp. 4057–4066.

[21] Maas, S. A., and Weiss, J. A., 2007, FEBio Theory Manual, http://mrl.sci.utah.edu/software/febio.

[22] Bonet, J., and Wood, R. D., 1997, Nonlinear Continuum Mechanics for Finite Element Analysis, Cambridge University Press, Cambridge, NY. [23] Matthies, H., and Strang, G., 1979, “The Solution of Nonlinear Finite Element Equations,” Int. J. Num. Methods Eng., 14, pp. 1613–1626.

[24] Bonet, J., and Wood, R. D., 1997, Nonlinear Continuum Mechanics for Finite Element Analysis, Cambridge University Press, Cambridge, NY.

[25] Simo, J. C., and Taylor, R. L., 1991, “Quasi-Incompressible Finite Elasticity in Principal Stretches: Continuum Basis and Numerical Algorithms,” Comput. Methods Appl. Mech. Eng., 85, pp. 273–310.

[26] Puso, M. A., and Solberg, J., 2006, “A Stabilized Nodally Integrated Tetrahedral,” Int. J. Num. Methods Eng., 67, pp. 841–867.

[27] Betsch, P., and Gruttmann, F., 1996, “A 4-Node Finite Shell Element for the Implementation of General Hyperelastic 3D-Elasticity at Finite Strains,” Comput. Methods Appl. Mech. Eng., 130, pp. 57–79.

[28] Mow, V. C., Kuei, S. C., Lai, W. M., and Armstrong, C. G., 1980, “Biphasic Creep and Stress Relaxation of Articular Cartilage in Compression: Theory and Experiments,” J. Biomech. Eng., 102, pp. 73–84.

[29] Ateshian, G. A., Ellis, B. J., and Weiss, J. A., 2007, “Equivalence Between Short-Time Biphasic and Incompressible Elastic Material Response,” J. Biomech. Eng. 129(3), pp. 405–412.

[30] Weiss, J. A., Maker, B. N., and Govindjee, S., 1996, “Finite Element Implementation of Incompressible, Transversely Isotropic Hyperelasticity,” Comput. Methods Appl. Mech. Eng., 135, pp. 107–128.

[31] Ateshian, G. A., Maas, S., and Weiss, J. A., 2010, “Finite Element Algorithm for Frictionless Contact of Porous Permeable Media Under Finite Deformation and Sliding,” J. Biomech. Eng., 132, p. 061006.

[32] Laursen, T. A., and Maker, B. N., 1995, “Augmented Lagrangian QuasiNewton Solver for Constrained Nonlinear Finite Element Applications,” Int. J. Numer. Methods Eng., 38, pp. 3571–3590.

[33] Maker, B. N., 1995, “Rigid Bodies for Metal Forming Analysis With NIKE3D,” University of California, Lawrence Livermore Laboratory, Report No. UCRLJC-119862, pp. 1–8.

[34] Schenk, O., and Gartner, K., 2004, “Solving Unsymmetric Sparse Systems of Linear Equations With PARDISO,” J. Future Gen. Comput. Syst., 20, pp. 475–487.

[35] Schenk, O., and Gartner, K., 2006, “On Fast Factorization Pivoting Methods for Symmetric Indefinite Systems,” Elect. Trans. Numer. Anal., 23, pp. 158–179. Available at http://etna.mcs.kent.edu/.

[36] Demmel, J. W., Eisenstat, S. C., Gilbert, J. R., Li, X. S., and Liu, J. W. H., 1999, “A Supernodal Approach to Sparse Partial Pivoting,” SIAM J. Matrix Anal. Appl., 20, pp. 720–755.

[37] Gupta, A., “WSMP: Watson Sparse Matrix Package,” IBM Research Report No. RC 21888 (98472), 2000.

[38] Veldhuizen, T., Jernigan, M., Ishikawa, Y., Oldehoeft, R., Reynders, J., and Tholburn, M., 1997, “Will C++ be Faster than FORTRAN? Scientific Computing in Object-Oriented Parallel Environments,” Lect. Notes Comput. Sci., 1343, pp. 49–56.

[39] Maker, B. N., 1995, “NIKE3D: A Nonlinear, Implicit, Three-Dimensional Finite Element Code for Solid and Structural Mechanics,” Lawrence Livermore Laboratory Technical Report No. UCRL-MA-105268.

[40] Park,S.,Krishnan,R.,Nicoll,S.B.,andAteshian, G.A.,2003,“CartilageInterstitial Fluid Load Supportin Unconfined Compression,” J.Biomech.,36,pp.1785–1796.

[41] Ogden, R. W., 1984, Non-Linear Elastic Deformations, Ellis Horwood, Chichester, UK.

[42] Erdemir, A., Viveiros, M. L., Ulbrecht, J. S., and Cavanagh, P. R., 2006, “An Inverse Finite-Element Model of Heel-Pad Indentation,” J. Biomech., 39, pp. 1279–1286.

[43] Ateshian, G. A., Warden, W. H., Kim, J. J., Grelsamer, R. P., and Mow, V. C., 1997, “Finite Deformation Biphasic Material Properties of Bovine Articular Cartilage From Confined Compression Experiments,” J. Biomech., 30, pp. 1157–1164.

[44] Holmes, M. H., 1986, “Finite Deformation of Soft Tissue: Analysis of a Mixture Model in Uni-Axial Compression,” J. Biomech. Eng., 108, pp. 372–381.

[45] Armstrong, C. G., Lai, W. M., and Mow, V. C., 1984, “An Analysis of the Unconfined Compression of Articular Cartilage,” J. Biomech. Eng., 106, pp. 165–173.

[46] Oden, J. T., 1972, Finite Elements of Nonlinear Continua, McGraw-Hill, New York.

[47] Treloar, L. R. G., 1944, “Stress-Strain Data for Vulcanised Rubber Under Various Types of Deformation,” Trans. Faraday Soc., 40, pp. 59–70.

[48] Bisshopp, R. E., and D. C. Drucker, 1945, “Large Deflection of Cantilever Beams,” Q. Appl. Math., 3, pp. 272–275.

[49] Batoz, J. L., 1982, “An Explicit Formulation for an Efficient Triangular Plate Bending Element,” Int. J. Numer. Methods Eng., 18, pp. 1077–1089.

[50] Moore, S. M., Ellis, B. J., Weiss, J. A., McMahon, P. J., and Debski, R. E., 2010, “The Glenohumeral Capsule Should be Evaluated as a Sheet of Fibrous Tissue: A Validated Finite Element Model,” Ann. Biomed. Eng., 38(1), pp. 66–76.

[51] Drury, N. J., Ellis, B. J., Weiss, J. A., McMahon, P. J., and Debski, R. E., 2011, “Finding Consistent Strain Distributions in the Glenohumeral Capsule Between Two Subjects: Implications for Physical Examinations,” J. Biomech., 44, pp. 607–613.

CHAPTER 3

A PLUGIN FRAMEWORK FOR EXTENDING THE SIMULATION CAPABILITIES OF FEBIO

3.1 Abstract

FEBio (Finite Elements for Biomechanics and Biophysics, www.febio.org) is a nonlinear finite element software suite that was developed for applications in the biomechanics and biophysics communities. FEBio employs mixture theory to account for the multiconstituent nature of biological materials, integrating the field equations for irreversible thermodynamics, nonlinear solid mechanics, fluid mechanics, mass transport with reactive species, and electrokinetics. This communication describes the development of a new “plugin” framework for FEBio. Plugins are dynamically linked libraries that allow users to add new features and to couple FEBio with other domain-specific software applications without modifying the source code directly. The governing equations and simulation capabilities of FEBio are reviewed. The implementation, structure, use and application of the plugin framework are detailed. Several example plugins are described in detail to illustrate how plugins enrich, extend, and leverage existing capabilities in FEBio, including applications to deformable image registration, prestrain in biological tissues, and coupling to an external software package that simulates angiogenesis using a discrete computational model. The plugin feature facilitates dissemination of new simulation methods, reproduction of published results, and coupling of FEBio with other domain-

specific simulation approaches such as compartmental modeling, agent-based modeling, and rigid body dynamics. We anticipate that the new plugin framework will greatly expand the application and thus the impact of the FEBio software suite.

3.2 Introduction

The Finite Element (FE) method has been used in the field of biomechanics as early as the 1970s (e.g. (Belytschko et al.; Davids and Mani; Doyle and Dobrin; Farah, Craig and Sikarskie; Janz and Grimm; Matthews and West)). However, the lack of FE software that is tailored to the needs of the field has hampered research progress and dissemination of models and results. Commercial packages that are most commonly used by the biomechanics community, such as Abaqus (Dassault Systèmes) and Ansys (Ansys, Inc.), are not specifically geared towards biological applications and are difficult to verify due to their closed source nature (Anderson, Ellis and Weiss; Henninger et al.). The limitations of commercial codes have forced many researchers to develop custom FE packages for solving problems in the biomechanics domain, such as (Laible et al.; Oomens, van Campen and Grootenboer; Simon et al.; Donzelli et al.; Sun et al.; Wayne, Woo and Kwan) to name only a few. However, few of these codes were actually made available in the public domain, which made verification and reproducibility of results challenging or impossible. Many of these codes were also not sufficiently general, but instead designed for solving very specific problems and did not support all features that are necessary for analyses in computational biomechanics and biophysics.

The FEBio project (Maas, Ellis, et al.), which was launched in 2007, was designed to overcome the lack of a general purpose finite element program for solving computational problems in biomechanics. It was geared towards the biomechanics and biophysics communities by providing functionality that is highly relevant in the field, such as accurate constitutive relations for mechanical, transport, and electrokinetic properties of tissues and cells; the ability to easily model anisotropy and material inhomogeneity; and the ability to prescribe boundary conditions and loading scenarios that would allow researchers to model the complex interactions between biological structures. The FEBio code is distributed as binaries (www.febio.org) for immediate use. The source code is also made available so that researchers can implement novel features with minimal effort. This greatly reduces the need for researchers to develop custom finite element codes from scratch.

The FEBio code was developed using the C++ programming language. Although many finite element codes, both commercial as well as open-source (e.g. Abaqus (Dassault Systèmes, Ansys (Ansys, Inc.) Nike3D (Lawrence Livermore National Lab), Feap (University of California, Berkeley), Code Aster (Électricité de France)) use the FORTRAN language, an object-oriented approach to programming is widely accepted as a superior programming model compared to procedural programming (Cary et al.). The C++ language was chosen since it is an object-oriented programming language and it has been shown that it performs comparable to FORTRAN (Veldhuizen et al.). Also, the possibility of developing efficient finite element codes based on object-oriented programming strategies has been demonstrated (Patzak and Bittnar; Zimmerman et al.). In addition, FEBio relies heavily on existing object-oriented design patterns (Gamma et al.; Reddy; Heng and Mackie) to resolve the many challenges in designing a complex finite element software library. In addition, the code is documented extensively: A Theory Manual details the theoretical foundations, a User's Manual explains the process of setting up and running FEBio models, and a Developer's Manual assists developers in implementing new features.

To further simplify the process of adding new features to FEBio, a plugin framework was developed that allows users to extend FEBio's standard feature set with minimal effort. Although users can download and modify the source code directly, building the entire FEBio code can be challenging, especially due to its dependency on third-party libraries. Even more, disseminating custom modifications to the source code may be challenging since the changes may be incompatible with future releases of FEBio. Plugins, which are essentially dynamically linked libraries, overcome these problems since they are developed independently from FEBio and can be linked at run-time. A plugin allows a user to add new features without having to build FEBio and solves the dissemination issue since plugins can be distributed independently of FEBio.

This communication presents FEBio's plugin framework. In Section 3.3, FEBio's modular framework is presented, as well as its kernel mechanism, which was central to the development of the plugin framework. In Section 3.4, several plugin examples are presented that illustrate the generality and versatility of the plugin framework, including a new material plugin that allows users to apply prestrain to their models, an implementation of the hyperelastic warping algorithm (Rabbitt et al.), a plugin that couples an agent-based modeling approach for simulating angiogenesis with FEBio. In Section 3.5, the advantages and disadvantages of the plugin framework are discussed and some suggestions for future improvements are provided as well.

3.3 Methods

3.3.1 Modular framework

FEBio is structured as a hierarchy of modules or libraries, each module collecting classes for solving a specific type of problem (Fig. 3.1). The FECore library is in many ways the “heart” of the code as it contains most of the base classes from which new features can be derived. It also contains the essential algorithms and data structures for solving FE problems. The FEBioMech library implements features related to solving nonlinear solid mechanics problems. The FEBioMix library builds upon the FEBioMech library by adding features that allow the solution of a coupled solid-fluid problems based on mixture theory. The NumCore library contains algorithms for solving linear systems of equations. Most of the classes in this library are proxy classes that serve as interfaces to external, third party libraries for solving linear systems of equations. The FEBioXML library reads the FEBio

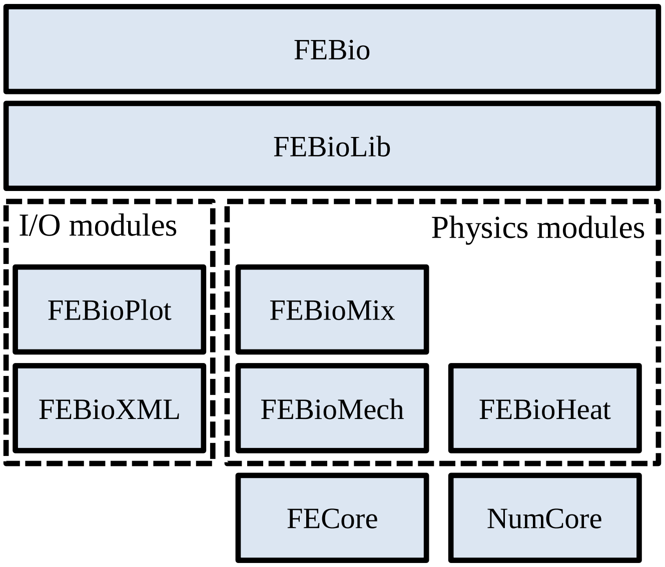


Fig. 3.1. Hierarchy of FEBio modules. The FECore library contains the kernel that manages all features. NumCore is a library that interacts with third party linear solvers. The physics modules implement physics-specific features. The I/O modules deal with input and output. The FEBioLib initializes the kernel class. The FEBio executable provides a command-line front end to the FEBio library.

input file and constructs a FE model that can be solved with FEBio. The FEBioPlot library writes the solution to a binary database, called the FEBio plot file.

The FECore module contains a special class, called the FECoreKernel. This kernel class contains a list of factory classes (Gamma et al.), which are used to instantiate the feature classes that implement specific functionality. A factory class is created and added to the kernel for each feature class. This process is termed feature registration and is done in the FEBioLib library. Each factory class also stores a class ID, which is a number that represents one of the predefined class categories (e.g. material, body load, plot field, etc.), and a string that identifies the feature class uniquely within its category. A particular feature class can now be instantiated by looking up the corresponding factory class using the class ID and string, and then requesting the factory class to create an instance of the feature class. One of the advantages of this approach is that it allows full automation of the I/O. The file FEBio parser, implemented in FEBioXML, is able to instantiate specific feature classes using only the information provided in the input file. Without this kernel mechanism, the FEBioXML library would have to be coupled to all the other libraries. It would also need to be modified each time a new feature is implemented. The kernel approach decouples the parser library from the other physics modules and automates the input. This is essential for the proper functioning of plugins since plugins cannot change the file parser code.

Although not specifically related to plugins, it is worth mentioning that FEBio has a substantial library of tensor classes. It offers various classes for representing first (i.e. vectors) second, third, fourth, fifth, and sixth order tensors and common operations that can be made with tensors (e.g. addition, multiplication, contraction, etc.). To maximize efficiency, different tensor symmetries are implemented in different classes. For example, the *mat3ds* class implements a second-order symmetric tensor, *mat3da* implements a skew-symmetric tensor, and *mat3dd* implements a diagonal second-order tensor. This library greatly facilitates the implementation of complex tensor expressions (Fig. 3.2).

3.3.2 Plugin mechanism

A plugin implements a new feature by deriving from one of the base classes that are defined in one of the modules. These base classes usually define an interface that derived classes must implement. For example, a new hyper-elastic constitutive material is implemented by deriving from the FEElasticMaterial class and implementing the *Stress* and *Tangent* functions. The plugin must also register the new feature class with the kernel by providing a factory class for the new feature class and passing this factory class to the kernel. FEBio offers various macros that hide the code details, which simplifies the

```
mat3ds FENeoHookean::Stress(FEMaterialPoint& mp)
{
   FEElasticMaterialPoint& pt =
                    *mp.ExtractData<FEElasticMaterialPoint>();
   double detF = pt.m_J;
   double detFi = 1.0/detF;
   double lndetF = log(detF);
   // calculate left Cauchy-Green tensor
   mat3ds b = pt.LeftCauchyGreen();
   // lame parameters
   double lam = m_v*m_E/((1+m_v)*(1-2*m_v));
   double mu  = 0.5*m_E/(1+m_v);
   // Identity
   mat3dd I(1);
   // calculate stress
   mat3ds s = (b - I)*(mu*detFi) + I*(lam*lndetF*detFi);
   return s;
}
```

Fig. 3.2. Stress evaluation for the neo-Hookean material. This figure illustrates the use of FEBio’s tensor class library which provides classes for representing tensors (mat3ds for symmetric second-order tensors, mat3dd for diagonal second-order tensors) and defines tensor operations. This library greatly simplifies the implementation of complicated tensor expressions.

registration process considerably.

Plugins are compiled and built independently from the main FEBio source code. However, in order for any plugin mechanism to work, there must be a communication mechanism set up that allows FEBio and the plugin to pass information to each other. This mechanism consists of two parts. First, FEBio and the plugin must ensure that they use the same data structures. This was accomplished by creating a Software Development Kit (SDK) that contains all the common data structures. This SDK essentially contains all the header files and prebuilt libraries that are used by FEBio and that the plugin could need. Second, the plugin must implement predefined functions that FEBio will call at specific points during its execution. For example, the PluginInitialize function gives the plugin a chance to do initialization, the PluginGetFactory function is used for registering a new feature class with the FEBio kernel. See the FEBio's developer documentation (www.febio.org) for more information about plugin development.

The specific classes and member functions that must be implemented depend on the particular type of plugin that is developed, or more specific, on the particular base class that the new feature class is derived from. In the next sections, several plugins types are explained in more detail. See the FEBio's developers manual for a more complete overview of the supported plugin types.

3.3.3 Plugins classes

3.3.3.1 Material plugin

A material plugin implements a new constitutive model. Constitutive models are used to evaluate solution-dependent quantities, such as stress, permeability, diffusivity, etc. A new constitutive model can be added by deriving a new class from a specific material base class. For instance, to implement a new constitutive model for elastic materials, a new class is derived from the FEElasticMaterial base class. The new class must implement the Stress (Cauchy stress) and Tangent (spatial elasticity tensor) functions. Implementation of the often complicated tensorial expressions for stress and elasticity tensors is greatly simplified using FEBio's extensive tensor class libraries (Fig. 3.2).

The prestrain plugin below shows an example of a material plugin. This plugin implements a general mechanism for applying prestrain to finite element models (Maas, Erdemir, et al.). The plugin implements a new material class that acts as a wrapper to an elastic material. Prestrain is modeled by defining a prestrain gradient tensor $\mathbf{F}_{ps}$ that is multiplied with the deformation gradient $\mathbf{F}$ and results in a total elastic deformation gradient,

$$\mathbf{F}_{eff} = \mathbf{F} \cdot \mathbf{F}_{ps} . \tag{3.1}$$

The prestrain material class evaluates this total elastic deformation gradient and then passes it to the child material class to evaluate the stress and elasticity tensors. This mechanism allows users to apply prestrain to any of FEBio's elastic material models. The plugin offers several mechanisms for generating the prestrain gradient and the applications range from enforcing an initial fiber stretch, to solving certain types of inverse FE models.

3.3.3.2 Nonlinear constraint plugin

Plugins can also implement nonlinear constraints that may constrain the solution of the problem. A nonlinear constraint can be used, for example, to maintain the distance between two nodes, to preserve the volume of an enclosed space, or to define a rigid joint between two rigid bodies. For this type of plugin, the base class is the FENLConstraint class, which defines three virtual functions that must be overwritten by the derived class. Two of these functions are involved in the evaluation of the equivalent nodal force vector and the stiffness matrix. Since FEBio uses an augmented Lagrangian approach for enforcing nonlinear constraints (Simo and Laursen), the third function, called Augment, updates the approximate Lagrange multipliers that enforce the constraint.

The FEWarp plugin below is an example of a nonlinear constraint plugin. Hyperelastic warping is an image registration method that is used to find the deformation, and resulting strains and stresses, in a finite element model by registering a template model to a target geometry (Rabbitt et al.). The image data is used directly to guide this registration process. An image energy functional is defined that measures the quality of the registration.

$$W_{HW}\left(\varphi\right)=\frac{\lambda}{2}\int_{V}\left(S\left(\varphi\left(\mathbf{X}\right)\right)-T\left(\mathbf{X}\right)\right)^{2}dV \tag{3.2}$$

Here, T is the template image, S is the target image, φ is the deformation map that maps a point from the reference to the deformed configuration, and the integral is evaluated over the FE domain. The warping algorithm tries to find the deformation that minimizes this functional. The advantage of this approach is that the deformation can be obtained without the specific knowledge of the applied loads to the model. Since the method is based on continuum mechanics, the resulting deformation is guaranteed to be a plausible, realistic deformation. The method has been applied successfully to characterize the mechanics of tissues, such as ligaments (Phatak et al.) and the left ventricle (Veress et al.). In the example below, the method will be applied on a simplified model of the left ventricle.

3.3.3.3 Plot field plugin

FEBio outputs the results of a simulation to a file that uses a self-describing, extendible format that allows users to customize its content to the specific needs of the analyses. The advantage of this approach is that the results can be stored in a compact file that only contains the data that is relevant for the analysis. FEBio offers a large set of output variables that users can choose from to customize the plot file. However, when adding a new feature, it is sometimes necessary to define new output variables that are not available from FEBio's default list of output variables. This can be accomplished by creating a plot-field plugin that defines the new output variables. The name assigned to the corresponding factory class is stored by FEBio to the output file so that postprocessors can identify the new data fields. Both the prestrain and the warping plugin examples below define new output variables.

3.3.3.4 Task plugin

A task plugin offers one of the most powerful abilities of the plugin framework. At the highest level, after parsing the input file, FEBio executes a single task. The default task is to solve the model defined by the FEBio input file, but other tasks can be executed as well. For instance, FEBio's optimization module is implemented as a task that calls the default solver task repeatedly while minimizing an objective function. In essence, a task defines the outer loop that controls what FEBio actually does. This is analogous to using FEBio as a library and implementing a main function that controls the program's execution. However, the task plugin approach has the advantage that all features of FEBio can be used, even those that otherwise would require linking to third-party libraries (e.g. linear solvers).

The AngioFE plugin (Edgar, Maas, et al.) shown below is an example of a task plugin that illustrates how FEBio can be linked to other libraries via the plugin framework. It uses a library (Edgar, Sibole, et al.) that simulates the growth, branching, and anastomosis of blood vessel sprouts in a collagen matrix. From experiments, it is known that the growing vessels deform the matrix in which they are embedded. In turn, it is believed that the deformation of the matrix directly affects the growth of the vessels. To couple the Angio library to FEBio, a task plugin was developed that allowed the two codes to interact. The task seeds the matrix with small vessel fragments using the Angio library and executes an initial growth step. The vessel fragments are grown using heuristics that control the growth speed, branching probability, and anastomosis. Then, the model is passed to FEBio which solves for the deformation of the matrix due to the growth of the vessel fragments. The interaction between the vessels and matrix is modeled as an elastic material that generates a contractile stress near the fragment tips. The new, deformed configuration is then passed back to the Angio library which then calculates the next growth step in the deformed model. The process continues until the vessels stop growing (Fig. 3.3).

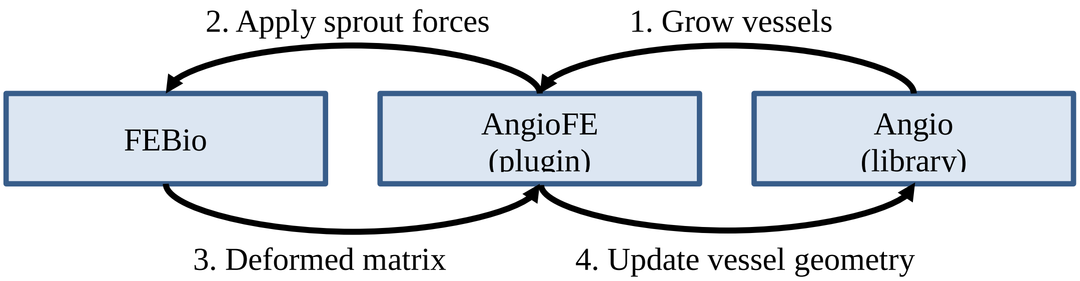


Fig. 3.3. The Angio plugin couples the Angio code, a library that simulates the growth of blood vessels, with FEBio. A two-way coupling is achieved by applying forces at the end of vessel fragments that deform the matrix. The deformed geometry in turn affects the growth of the blood vessel fragments.

3.4 Results

3.4.1 Prestrain plugin

The prestrain plugin was used to compare the effective (von Mises) stress predictions of two dog-bone models representing ligament tissue (Fig. 3.4). The constitutive model was assumed to be transversely isotropic Mooney-Rivlin (Weiss, Maker and Govindjee) and the fibers ran along the long axis of the geometry. In one of the models, the prestrain plugin was used to apply an initial 10% fiber stretch. Due to symmetry, only half of the model was considered in the analysis.

3.4.2 FEWarp

The warping plugin was applied to a 3D model of a human left ventricle (Fig. 3.5). The FE model was created from segmenting the left ventricle from the template image data

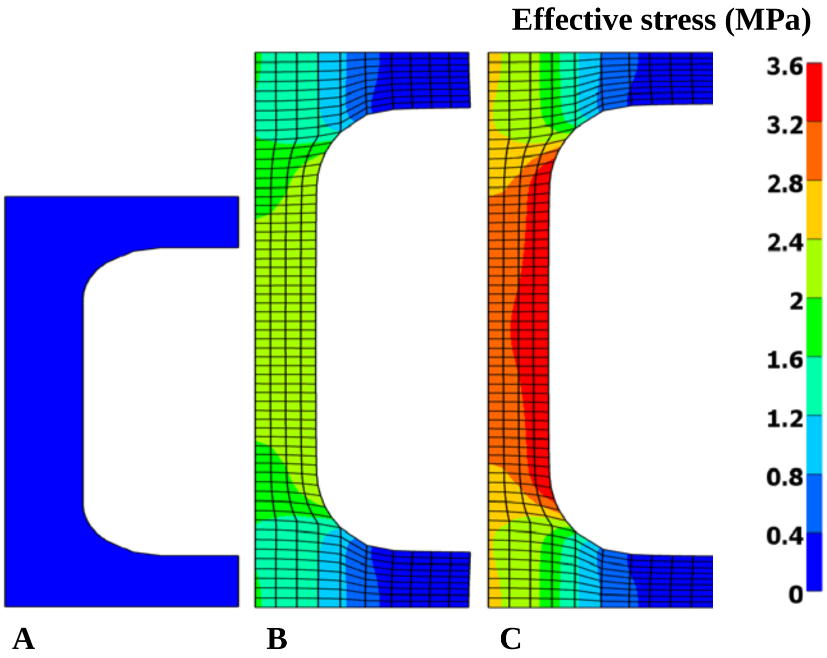


Fig. 3.4. Prestrain analysis of dogbone sample, using a transversely isotropic material. The fibers run along the long axis of the geometry. A) Original geometry of half-symmetry model. B) Effective (von-Mises) stress without prestrain. C) Effective (von-Mises) stress with 10 percent fiber stretch applied to the fibers.

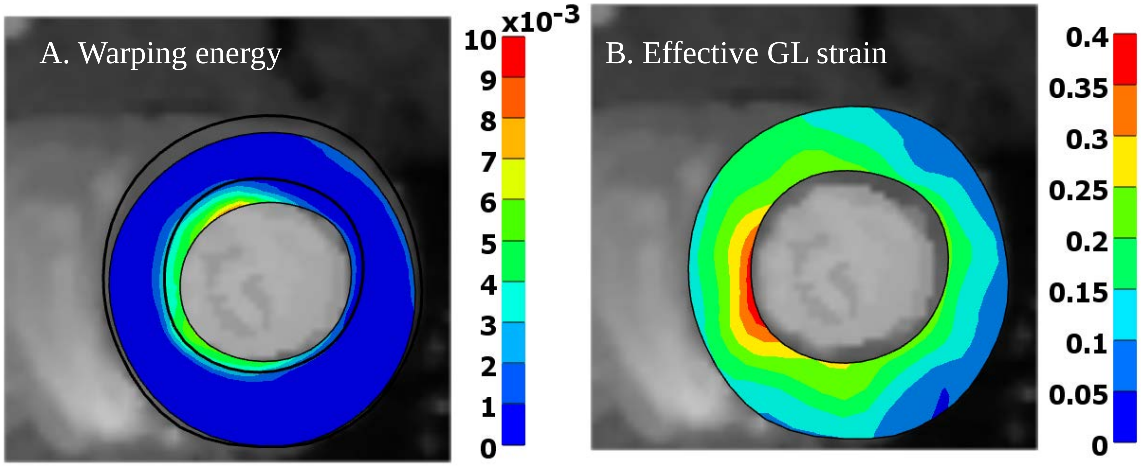


Fig. 3.5. 2D slice from a 3D warping analysis that tracks the deformation of the left ventricle. A) Warping energy, measuring the mismatch between the template and the target image. The black contour shows the outline of the target geometry. B) Effective Green Lagrange strain in the final, deformed state.

set. The target image data set was created by simulating the effect of inflation of the left ventricle under a constant pressure. The specific process how this was accomplished was detailed in Veress et al. and since the solution to the warping problem was known in this case, the approach can be used to verify the warping algorithm. The plugin was then used to obtain the deformation from template to target image data. This plugin also offers several new output variables, including the template and target image data, mapped onto the FE mesh, and the warping energy, which can be visualized to identify the initial registration error (Fig. 3.5, panel A). After successful registration, the strain can be plotted to inspect the deformation (Fig. 3.5, panel B).

3.4.2 AngioFE

The AngioFE plugin (Edgar, Maas, et al.) was used to simulate the growth of sprout vessels in a 3D rectangular matrix (Fig. 3.6). The mesh of the matrix was only anchored at some of the corners to simulate the deformation of a free-floating gel. The matrix was

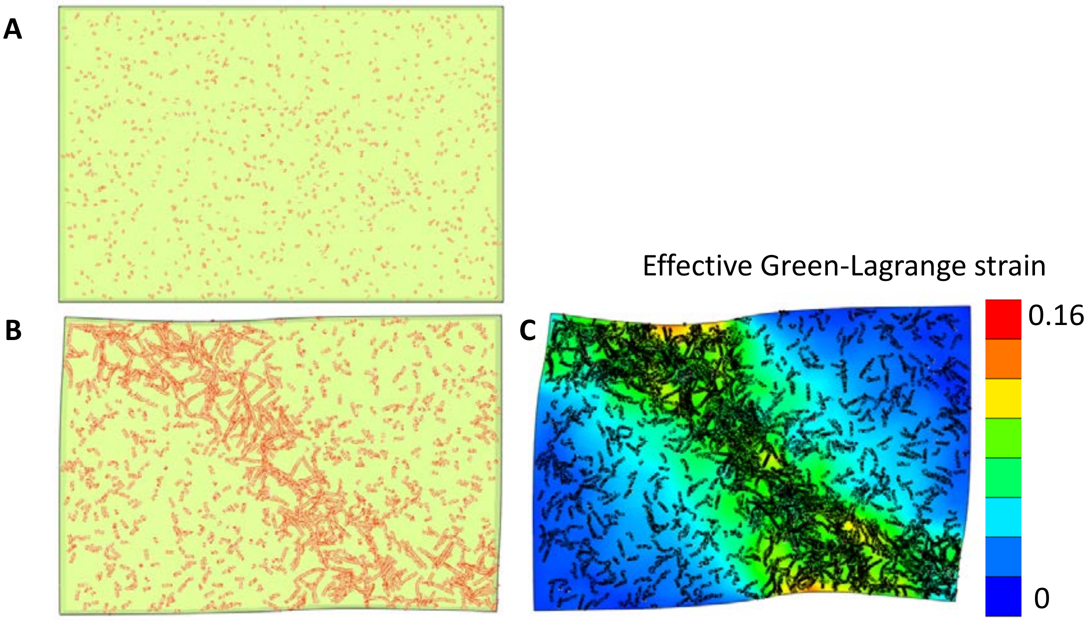


Fig. 3.6. Simulation of angiogenesis. A) Initial geometry and vessel fragments. B) Vessel network at intermediate time step. Some initial deformation is noticeable. C) Final geometry and vessel network. The effective Green-Lagrange strain is overlaid to show the areas of largest deformation.

modeled using a visco-hyper-elastic material model. The initial fragments were seeded randomly in the volume of the matrix (Fig. 3.6, panel A). The collagen density, which in part drives the vessel growth, was lowered in a narrow band around the diagonal of the mesh compared to the rest of the model. As a result, the vessel fragments in this narrow band grew faster and branched more often, generating a denser network of vessels in this area of the mesh (Fig. 3.6, panel B). At the end of the simulation, the matrix shows significant deformation and the resulting strains are larger in the area where the vessel network is denser (Fig. 3.6, panel C).

3.5 Discussion

This communication presented a plugin framework for FEBio, which allows users to easily develop custom features with minimal effort and without the need to change the

FEBio source code directly. Using plugins, users can add new constitutive models, loads, nonlinear constraints, plot data fields, tasks, etc., and thus tailor FEBio to their specific research needs. Since user plugins are developed and maintained independently from the FEBio source code, it is easier to develop and share novel features. This also simplifies the reproducibility of results obtained using these plugins, since end-users only need to acquire the plugin and do not need to rebuild the entire FEBio source code. To facilitate the sharing of new user plugins even more, the FEBio project offers a webpage that hosts plugins (febio.org/plugins). Users can download FEBio plugins from this website and also submit their own plugins. All the plugin examples in this article are available from this website.

Scripting is another common mechanism that many software packages support for custom extensions. Scripting is comparable to plugins in that it allows users to extend the basic feature set, but differs in some important aspects. First of all, a script does not need to be compiled and can be used across multiple platforms. This is a distinct advantage over plugins, which usually need to be built separately on each platform. However, scripts may suffer from a performance penalty because they need to be interpreted while the script is running. For time intensive processes, such as evaluation of a material’s stress and elasticity tensors, a plugin may offer noticeable performance benefits. Nevertheless, both plugins and scripts have their value, and for this reason, there are plans to add a scripting functionality to FEBio.

It is worth mentioning that other, related software packages also offer some mechanisms for extending its core set of functionalities. For instance, Abaqus (Dassault Systèmes) and Ansys (Ansys, Inc.), both commercial finite element software that is commonly used in biomechanics, allow users to write so-called user subroutines. Using these user subroutines, users can add new features, such as material subroutines, etc. Regarding the addition of new material models, a recent publication (Pierrat et al.) compared Abaqus' approach to FEBio's and concluded that FEBio's approach is superior in several ways. In particular, the use of C++ and FEBio's tensor class library was seen as a great advantage over Abaqus's approach, which requires programming in FORTRAN. One advantage Abaqus' approach has is that the dynamic libraries are built behind the scene. Abaqus calls a FORTRAN compiler that builds the library and then links it automatically to the executable. Aside from an initial setup, the user is not involved in the build process. A similar mechanism is planned for FEBio that will simplify writing plugins even more.

As the examples in this article have illustrated, FEBio's plugin mechanism offers a powerful way for adding new capabilities to FEBio. Using this framework, researchers have an almost unlimited tool at their disposal to tailor FEBio to their specific needs. Although the source code of FEBio will remain freely available to the research community, it is hoped that this new plugin framework will be embraced as the preferred approach for extending FEBio.

3.6 References

Anderson, A. E., B. J. Ellis, and J. A. Weiss. "Verification, Validation and Sensitivity Studies in Computational Biomechanics." *Comput. Methods Biomech. Biomed. Eng.* 10.3 (2007).

Belytschko, T., et al. "Finite Element Stress Analysis of an Intervertebral Disc." *J. Biomech.* 7.3 (1974): 277-85.

Cary, J.R., et al. "Comparison of C++ and Fortran 90 for Object-Oriented Scientific Programming." *Comput. Phys. Commun.* 105.1 (1998): 20-36.

Davids, N., and M. K. Mani. "A Finite Element Analysis of Endothelial Shear Stress for

Pulsatile Blood Flow." *Biorheology* 11.2 (1974): 137-47.

Donzelli, P. S., et al. "Automated Adaptive Analysis of the Biphasic Equations for Soft Tissue Mechanics Using a Posteriori Error Indicators." *Intl. J. Num. Meth. Eng.* 34.3 (1992): 1015-33.

Doyle, J. M., and P. B. Dobrin. "Finite Deformation Analysis of the Relaxed and Contracted Dog Carotid Artery." *Microvasc. Res.* 3.4 (1971): 400-15.

Edgar, Lowell T., et al. "A Coupled Model of Neovessel Growth and Matrix Mechanics Describes and Predicts Angiogenesis in Vitro." *Biomech. Model. Mechanobiol* 14.4 (2015): 767-82.

Edgar, Lowell T., et al. "A Computational Model of in Vitro Angiogenesis Based on Extracellular Matrix Fiber Orientation." *Comput. Method. Biomec.* 16.7 (2013).

Farah, J. W., R. G. Craig, and D. L. Sikarskie. "Photoelastic and Finite Element Stress Analysis of a Restored Axisymmetric First Molar." *J. Biomech.* 6.5 (1973): 511-20.

Gamma, Erich, et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995. Print.

Heng, B. C. P., and R. I. Mackie. "Using Design Patterns in Object-Oriented Finite Element Programming." *Comput. Struct.* 87.15–16 (2009): 952-61.

Henninger, H. B., et al. "Validation of Computational Models in Biomechanics." *Proc. Inst. Mech. Eng. H* 224.7 (2010): 801-12.

Janz, R. F., and A. F. Grimm. "Finite-Element Model for the Mechanical Behavior of the Left Ventricle. Prediction of Deformation in the Potassium-Arrested Rat Heart." *Circ. Res.* 30.2 (1972): 244-52.

Laible, J. P., et al. "A Poroelastic-Swelling Finite Element Model with Application to the Intervertebral Disc." *Spine* 18.5 (1993): 659-70.

Maas, Steve A., et al. "Febio: Finite Elements for Biomechanics." *Journal of Biomechanical Engineering* 134.1 (2012).

Maas, Steve A., et al. "A General Framework for Application of Prestrain to Computational Models of Biological Materials." *Journal of the Mechanical Behavior of Biomedical Materials* 61 (2016): 499-510.

Matthews, F. L., and J. B. West. "Finite Element Displacement Analysis of a Lung." *J. Biomech.* 5.6 (1972): 591-600.

Oomens, C. W., D. H. van Campen, and H. J. Grootenboer. "A Mixture Approach to the Mechanics of Skin." *J. Biomech.* 20.9 (1987): 877-85.

Patzak, B., and Z. Bittnar. "Design of Object Oriented Finite Element Code." *Adv. Eng. Softw.* 32 (2001): 759-67.

Phatak, N. S., et al. "Strain Measurement in the Left Ventricle During Systole with Deformable Image Registration." *Med. Image Anal.* 13.2 (2009): 354-61.

Pierrat, Baptiste, et al. "Finite Element Implementation of a New Model of Slight Compressibility for Transversely Isotropic Materials." *Comput. Method. Biomec.* 19.7 (2015): 1-14.

Rabbitt, R. D., et al. "Mapping of Hyperelastic Deformable Templates." *Proc. SPIE Int. Soc. Opt. Eng.* 2552 (1995): 252-64.

Reddy, Martin. *Api Design for C++*. Elsevier, Inc, 2011. Print.

Simo, J. C., and T. A. Laursen. "Augmented Lagrangian Treatment of Contact Problems Involving Friction." *Computers and Structures* 42.1 (1992): 97-116.

Simon, B. R., et al. "A Poroelastic Finite Element Formulation Including Transport and Swelling in Soft Tissue Structures." *J. Biomech. Eng.* 118.1 (1996): 1-9.

Sun, D. N., et al. "A Mixed Finite Element Formulation of Triphasic Mechano-Electrochemical Theory for Charged, Hydrated Biological Soft Tissues." *Intl. J. Num. Meth. Eng.* 45.10 (1999): 1375-402.

Veldhuizen, T., et al. "Will C++ Be Faster Than Fortran? Scientific Computing in Object-Oriented Parallel Environments." Vol. 1343. Lecture Notes in Computer Science: Springer Berlin / Heidelberg, 1997. 49-56. Print.

Veress, Alex I., et al. "Measuring Regional Changes in the Diastolic Deformation of the Left Ventricle of Shr Rats Using Micropet Technology and Hyperelastic Warping." *Ann. Biomed. Eng.* 36.7 (2008): 1104-17.

Wayne, J. S., S. L. Woo, and M. K. Kwan. "Application of the U-P Finite Element Method to the Study of Articular Cartilage." *J. Biomech. Eng.* 113.4 (1991): 397-403.

Weiss, J.A., B.N. Maker, and S. Govindjee. "Finite Element Implementation of Incompressible, Transversely Isotropic Hyperelasticity." *Comput. Method. Appl. M.* 135 (1996): 107-28.

Zimmerman, Th., et al. "Aspects of an Object-Oriented Finite Element Environement." *Comput. Struct.* 68 (1998).

CHAPTER 4[1]

FINITE ELEMENT SIMULATION OF ARTICULAR CONTACT MECHANICS WITH QUADRATIC TETRAHEDRAL ELEMENTS

4.1 Abstract

Although it is easier to generate finite element discretizations with tetrahedral elements, trilinear hexahedral (HEX8) elements are nearly always used in simulations of articular contact mechanics. This is due to numerical shortcomings of linear tetrahedral (TET4) elements, limited availability of quadratic tetrahedron elements in combination with effective contact algorithms, and the perceived increased computational expense of quadratic finite elements. In this study we implemented both ten-node (TET10) and fifteen-node (TET15) quadratic tetrahedral elements in FEBio (www.febio.org) and compared their accuracy, robustness in terms of convergence behavior and computational cost for simulations relevant to articular contact mechanics. Suitable volume integration and surface integration rules were determined by comparing the results of several benchmark contact problems. The results demonstrated that the surface integration rule used to evaluate the contact integrals for quadratic elements affected both convergence

[1] Reprinted from *Journal of Biomechanics*, Vol. 49, Issue 5. Steve A. Maas, Benjamin J. Ellis, David S. Rawlins, Jeffrey A. Weiss, “Finite element simulation of articular contact mechanics with quadratic tetrahedral elements”, with permission from ELSEVIER.

behavior and accuracy of predicted stresses. The computational expense and robustness of both quadratic tetrahedral formulations compared favorably to the HEX8 models. Of note, the TET15 element demonstrated superior convergence behavior and lower computational cost than both the TET10 and HEX8 elements for meshes with similar numbers of degrees of freedom in the contact problems that we examined. Finally, the excellent accuracy and relative efficiency of these quadratic tetrahedral elements was illustrated by comparing their predictions with those for a HEX8 mesh for simulation of articular contact in a fully validated model of the hip. These results demonstrate that TET10 and TET15 elements provide viable alternatives to HEX8 elements for simulation of articular contact mechanics.

4.2 Introduction

Advances in imaging and computational methods make it possible to create and analyze detailed subject-specific models of biomechanical structures from high resolution image data. In this paper we focus on subject-specific finite element (FE) analysis of articular joint contact mechanics. Both generic and subject-specific models of articular contact have been developed and validated to gain insight into load transfer, cartilage mechanics and the etiology of osteoarthritis in the knee (Khoshgoftar et al.; Luczkiewicz et al.), hip (Harris et al.; Henak et al.) ankle (Anderson et al.; Kern and Anderson) and spine (Dreischarf et al.; Von Forell et al.), among other joints. Despite the progress that has been made in modeling subject-specific joint contact mechanics, many challenges still remain. The articular cartilage of most joints in the human body has complex geometry, undergoes large deformations, is subjected to large compressive loads, and is often thin compared to the surrounding anatomical support. These challenges make it difficult to obtain accurate, validated computational models of articular contact mechanics.

The element type used to discretize the articular geometry is one of the most important choices that affects accuracy and robustness in simulations of articular contact. Linear tetrahedral (TET4) elements are often used due to the ease and robustness of performing automatic meshing, and local and adaptive refinement with tetrahedral elements (Hubsch et al.; Johnson and MacLeod; Prakash and Ethier; Donzelli et al.) (Delaunay; Wrazidlo et al.; Lo; Lo; Lohner; Shephard and Georges; McErlain et al.). There are several examples in the recent literature that have used TET4 elements to discretize articular cartilage (Das Neves Borges et al.; Johnson et al.; McErlain et al.). However, the TET4 element has several well-known numerical issues. First, TET4 elements can only represent a constant strain state, which necessitates a very fine discretization, often requiring long solution times. Second, TET4 elements lock for nearly incompressible materials as well as under bending deformations (Hughes), which further reduces their accuracy. Because of these issues, trilinear hexahedral elements (HEX8) have seen much wider use in joint contact analyses, despite the fact that creating hexahedral meshes for complex geometries can be challenging and time consuming. Alternative formulations for TET4 elements have been designed to circumvent their problems (e.g., (Puso and Solberg; Gee et al.) based on nodal averaging of the deformation gradient). Although they reduce locking, they have other problems such us spurious deformation modes (Maas, Ellis, Rawlins, et al.), which make them inaccurate for contact analysis.

Quadratic tetrahedral elements are an attractive alternative to TET4 elements. They maintain the advantages of tetrahedral mesh generation and they can represent curved boundaries more accurately than HEX8 elements since their edges and faces can deform. Although quadratic tetrahedral elements have been investigated as alternatives to HEX8 elements (e.g., (Tadepalli, Erdemir and Cavanagh; Cifuentes and Kalbag; Weingarten)), none of these studies investigated their application for simulation of articular contact analyses. The 10-node tetrahedron (TET10) has seen some limited use for contact mechanics (Tadepalli, Erdemir and Cavanagh; Yang and Spilker; Bunbar et al.; Wan, Hoa and Wen; Hao et al.). To our knowledge the 15-node tetrahedron (TET15) has never been used in computational biomechanics and it has seen very limited use in nonlinear computational solid mechanics at all (Danielson), although it has been used in the fluid mechanics community (Bertrand, Gadbois and Tanguy).

The objectives of this study were to determine the efficacy in terms of accuracy of the recovered stresses, robustness in terms of the convergence behavior, and computational expense of the TET10 and TET15 elements compared to the HEX8 element in the context of articular contact mechanics. First, the effects of different integration rules on the stress predictions and computational cost were investigated and used to determine suitable integration rules for both the TET10 and TET15 elements. Then using these integration rules, the accuracy and computational cost of both elements were compared to the HEX8 element for several benchmark contact problems. For one of these problems, we examined the results obtained using TET4 elements to contrast their performance with the quadratic elements. Finally, we compared predictions of contact stresses using HEX8, TET10, and TET15 elements for a validated model of hip contact mechanics (Henak et al.).

4.3 Methods

4.3.1 Element formulation and numerical integration

All element formulations in this research were implemented in the FEBio software suite (www.febio.org), which uses an implicit, Newton-based method to solve the nonlinear FE equations of solid mechanics (Maas, Ellis, Ateshian, et al.). The TET10 element has 10 nodes: 4 corner nodes and 6 nodes located at the midpoint of the edges (Fig. 4.1). Due to the quadratic shape functions, the facets and edges of this element can distort and therefore the element behavior is “softer” than the TET4 element. The TET15 element adds one more node at the center of each facet, and one in the center of the tetrahedron (Fig 4.1). Although the TET15 element has more nodes than the TET10, it is still a quadratic element since the highest order of complete polynomial that can be represented by the shape functions is second order. The TET15 element can represent some forms of quadratic strains, whereas the TET10 element can only represent linear strains.

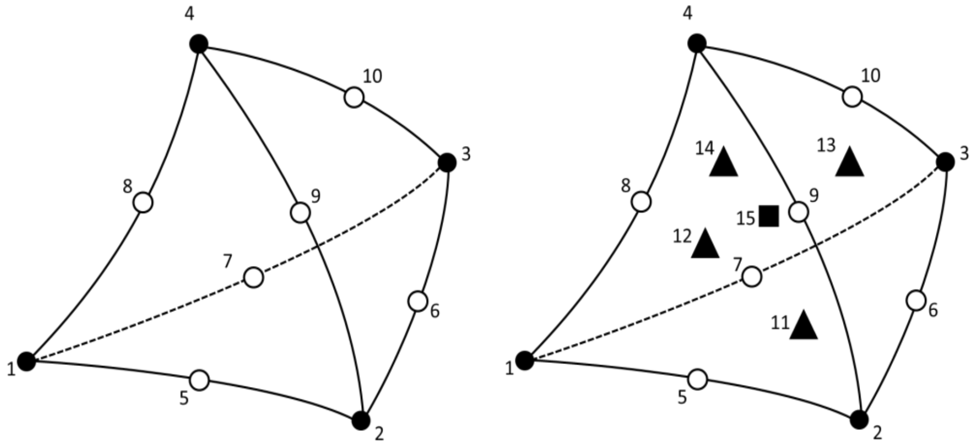


Fig. 4.1: Schematic of the node topology for two quadratic tetrahedral elements that were examined in this study. Left, 10-node quadratic tetrahedron (TET10). Right, 15-node quadratic tetrahedron (TET15). Closed circles represent corner nodes, open circles represent edge nodes, triangles represent facet center nodes, and the square in the right image represents the center node of the TET15 element.

In a FE formulation, the discretized form of the equilibrium equations requires the use of appropriate numerical integration schemes. For second-order elements, the integration rule should have at least second-order accuracy, at least for linear analyses. The ideal integration rule is less clear for large deformation nonlinear analyses. For this reason, we implemented and compared several volumetric integration rules. We denote volume integration rules as V(n) where (n) indicates the number of integration points. Similarly, the notation S(n) is used to denote surface integration rules. For the TET10 element, both 4-point (V4) and 8-point (V8) Gauss integration rules were implemented (Abramowitz and Stegun). For the TET15 element, 11-point (V11) and 15-point (V15) Gauss integration rules were implemented (Keast). All of these rules are at least second-order accurate (for linear analyses) and are symmetric, i.e. the integration points are distributed in a symmetrical spatial pattern.

For contact enforcement, a surface integration rule is required to integrate the traction forces over the discrete surface, represented by facets of the finite elements. The facets of the TET10 element are 6-node quadratic triangles. Two surface integration rules were implemented and compared: 3-point Gauss (S3), and 7-point Gauss (S7) (Fig. 4.2). For the TET15 element, a 7-node quadratic facet, and the same S3 and S7 integration rules were implemented. Again, these rules have at least second-order accuracy and are symmetric (Abramowitz and Stegun).

The mean dilatation formulation was used for the HEX8 models (Simo and Taylor). This formulation is known to perform well for nearly-incompressible materials for which the standard displacement-based formulation, using full integration, has a tendency to lock. All analyses were performed using the quasi-Newton solver in FEBio (Maas, Rawlins, et

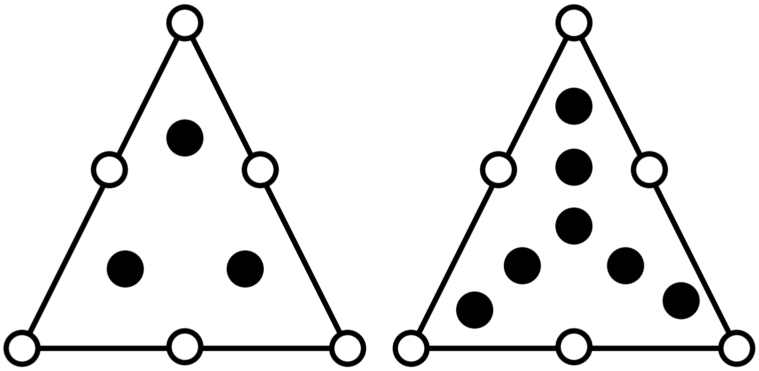


Fig. 4.2: Schematic of the surface integration rules that were implemented and tested for evaluation of the contact integrals for the quadratic triangle facets composing the surface of the tetrahedral elements. Left panel: S3 integration rule. Right panel: S7 integration rule. Open circles represent nodes, filled circles represent integration points. Note that these integration rules are symmetric in spatial distribution.

al.), which is based on the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method (Matthies and Strang). This nonlinear solver begins with a full formation and factorization of the stiffness matrix and then proceeds with a user-defined number of BFGS updates, which involve computation of the right-hand-side (RHS) vector. For two of the problems below, we report both the number of stiffness matrix reformations and RHS evaluations as metrics of nonlinear convergence and computational effort. All contact analyses used the “sliding-tension-compression” contact algorithm in FEBio (Maas, Rawlins, et al.). This algorithm implements a facet-on-facet, frictionless sliding contact where the contacting surfaces can separate but not penetrate (Laursen). The augmented Lagrangian method was used to enforce the contact constraint to a user-defined tolerance (Laursen and Maker).

4.3.1 Stress recovery

During the FE solution process, stresses are typically evaluated at the integration points. Since stresses are calculated using the shape function derivatives, they are usually discontinuous across element boundaries. For visualization and for further post-processing analysis (e.g. a-posteriori stress error estimates (Zienkiewicz and Zhu)), a more accurate, continuous stress field is required. This is accomplished using a stress-recovery algorithm that projects the stresses from the integration points to the nodes. For TET4 elements, a weighed nodal average of the surrounding element values is usually sufficient, but for quadratic elements this does not produce good results. We implemented and used the Superconvergent Patch Recovery (SPR) method to recover stresses for the quadratic elements (Zienkiewicz and Zhu). This method incorporates a least-squares approach to fit a polynomial to the integration point values of an element patch surrounding each node. The polynomial is then evaluated at the nodes to determine the nodal value. Another important reason for investigating different integration rules is to determine the minimal integration rule that recovers the stresses accurately. Since the SPR method uses least-squares, each node must be surrounded by a minimal number of integration points to solve the least-squares system. This could pose problems at the boundary of the mesh and in thin structures. For the TET10 element, we used a linear least-squares fit since there are often not enough integration points (especially for the V4 integration rule) to fit a complete quadratic polynomial. For the TET15 element, a complete quadratic polynomial was used.

4.3.2 Plane strain contact

Different combinations of surface and volume integration rules were investigated using a plane strain contact model (Fig. 4.3). This model is representative of articular contact mechanics in that it models the contact between curved, thin, incongruent, deformable materials. The model consisted of two parts. The top part had dimensions of 1.5 mm (height) x 20 mm (length) x 1 mm (thickness). The bottom part had the same height and thickness but was curved downward with a radius of curvature of 58 mm. The bottom

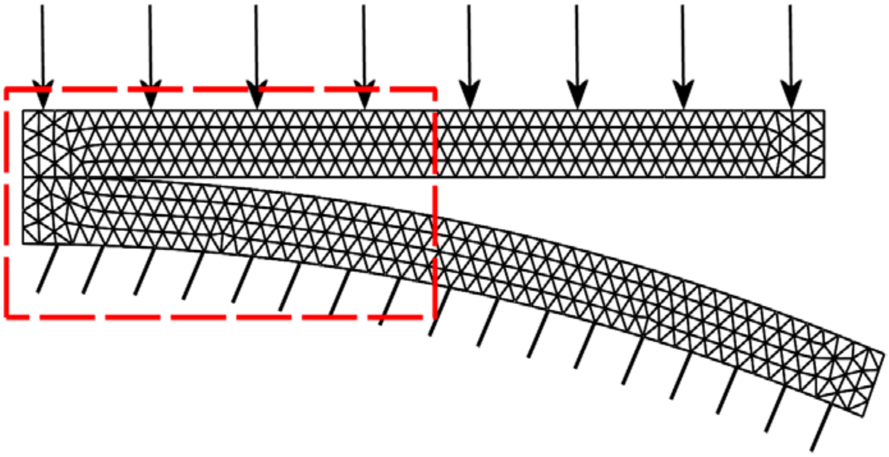


Fig. 4.3: Schematic of the plane strain contact problem. The bottom surface of the curved layer is held fixed, while the top layer undergoes a prescribed downward displacement. Dashed area shows the region illustrated in the contour plots in Fig. 4.5.

of the curved layer was fixed while the top of the flat layer was displaced downward. Since FEBio is designed for 3D analysis, plane strain boundary conditions were created by preventing the nodes from displacing in the out-of-plane direction. The materials of both layers were represented by an uncoupled Mooney-Rivlin constitutive model with strain energy:

$$W_{MR} = c_1\left(\tilde{I}_1 - 3\right) + c_2\left(\tilde{I}_2 - 3\right) + \frac{1}{2}k\left(\ln J\right)^2 . \tag{4.1}$$

Here, $\tilde{I}_1$, $\tilde{I}_2$ are the first and second invariants of the deviatoric right Cauchy-Green deformation tensor, and J is the determinant of the deformation gradient. This material formulation and the material coefficients below were chosen to represent articular and femoral articular cartilage and coincide with the properties of a validated hip joint model (discussed below). The material parameters were c_1 = 2.98 MPa, c_2 = 0 MPa, k = 594 MPa, for the flat layer; and c_1 = 1.86 MPa, c_2 = 0 MPa, k = 371 MPa for the curved layer. To compare the integration rules, we examined isovalue fringe plots of the 3rd principal stress and searched for erroneous patterns in the isovalues.

4.3.3 Compression and sliding of a rigid cylinder on a deformable block

To assess the performance of the quadratic tetrahedral elements in contact problems with large amounts of sliding, a benchmark problem from the FEBio Verification Suite was used (co07.feb). This is a revelant test problem since many joints (e.g. knee, shoulder, hip) undergo large amounts of sliding during articular contact. A rigid cylinder (outer radius = 1 mm) first indents a rectangular block (length = 5 mm, width = 1.9 mm, height = 1 mm) and then slides across the surface of the block (Fig. 4.4). The cylinder was modeled as a rigid body using quadratic 20-node hexahedral elements. Mid-side nodes of the elements were positioned so that the surface of the cylinder was smooth to avoid jumps in the contact stiffness due to discontinuities in the surface normal. Again, as before, the material formulation was chosen to represent articular cartilage. The rectangular block was represented using the Mooney-Rivlin model above (c_1 = 2.98 MPa, c_2 = 0 MPa, k = 594 MPa) and was discretized with TET4, TET10 and TET15 elements. A model using HEX8 elements was created and used as a reference. A mesh convergence study was conducted to allow fair assessment of the computational costs of the four element types. Convergence was measured in terms of average reaction force on the rigid indenter over the entire horizontal displacement of the indenter. Fringe plots of the 3rd principal stress at the midpoint of the analysis were used to compare the predictions of the models.

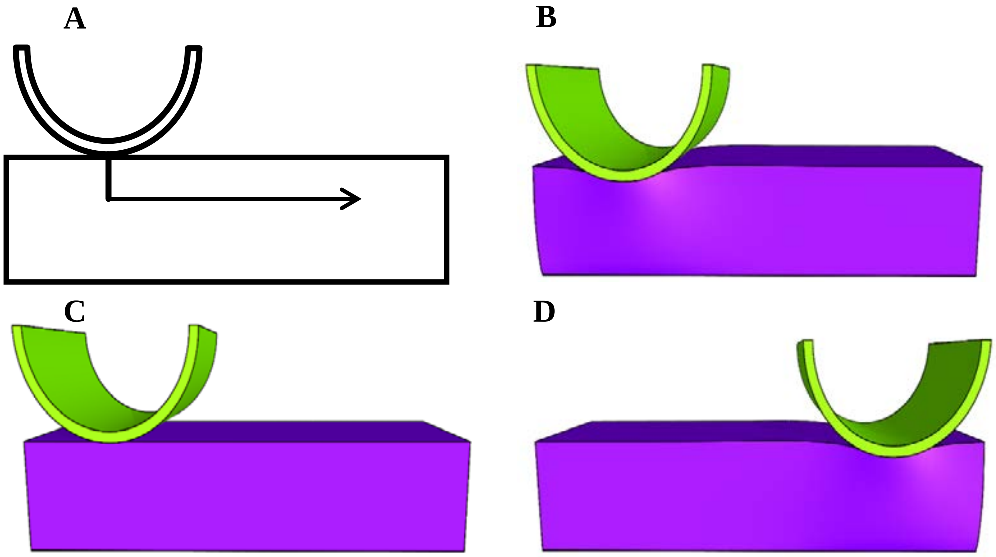


Fig. 4.4: Geometry and loading path of the rigid cylinder on the deformable box. First, the cylinder was displaced vertically. After reaching maximal vertical displacement, a horizontal displacement was applied while the vertical position was maintained. From top to bottom: schematic diagram, initial configuration, configuration at maximal vertical displacement, final configuration.

4.3.4 Articular contact in the human hip

To test the effectiveness of quadratic tetrahedral elements for production size articular contact problems involving complex geometry, the articular layers in a model of the hip from a previous validation study (Henak et al.) were discretized with tetrahedral elements. In this previous study, articular layers were discretized with HEX8 elements and after a mesh convergence study it was determined that six elements through the thickness produced converged results. The thickness varied across the articular surfaces and ranged from 0.2 mm to 3.7 mm on the femoral side and from 0.5 mm to 4.5 mm on the acetabular side. The HEX8 model was validated with experimental measurements of contact stress. Cartilage layers were modeled using a Mooney-Rivlin material (acetabular cartilage c_1=2.98 MPa, c_2=0 MPa, k=594 MPa, femoral cartilage c_1=1.86 MPa, c_2=0 MPa, k=371

MPa). For this study, two additional models were created using TET10 and TET15 elements. The mesh resolution was chosen such that the number of nodes (and thus the degrees of freedom) roughly matched the HEX8 model. For the tetrahedral models the number of elements through the thickness varied from one to four depending on the cartilage thickness. In the areas where contact was occurring there were about two to three elements through the thickness on average. Loading conditions were applied that simulated walking heel strike (Bergmann et al.). A prescribed load of 1650 N was applied to the femur while keeping the pelvis fixed. The 3rd principal stress, an indicator of contact stress, and contact area were compared.

4.4 Results

4.4.1 Plane strain contact

The surface integration rule had a large effect on predictions of 3rd principal stress (Fig. 4.5). The S3 surface integration rule predicted overall larger variations in stress near the contact interface than the S7 rule (compare rows 2 and 3 in Fig. 4.5). We concluded that the S7 surface integration rule is the best choice of the two rules. In contrast, there was little effect of volume integration rule on predicted stresses (compare rows 3 and 4 in Fig. 4.5). Indeed, comparing for instance the combination S3/V4 with S3/V8 for the TET10 element (Fig. 4.5, left column), the results are nearly indistinguishable. A noticeable exception is the combination S3/V11 and S3/V15 for the TET15 element where the stresses do show a significant difference. We found that the combination S3/V11 ran very poorly and we suspect that this combination is unstable and should be avoided. Although the other results show little variation for the different volume integration rules, we did observe improved convergence behavior for the higher-order volume integration rules. For this

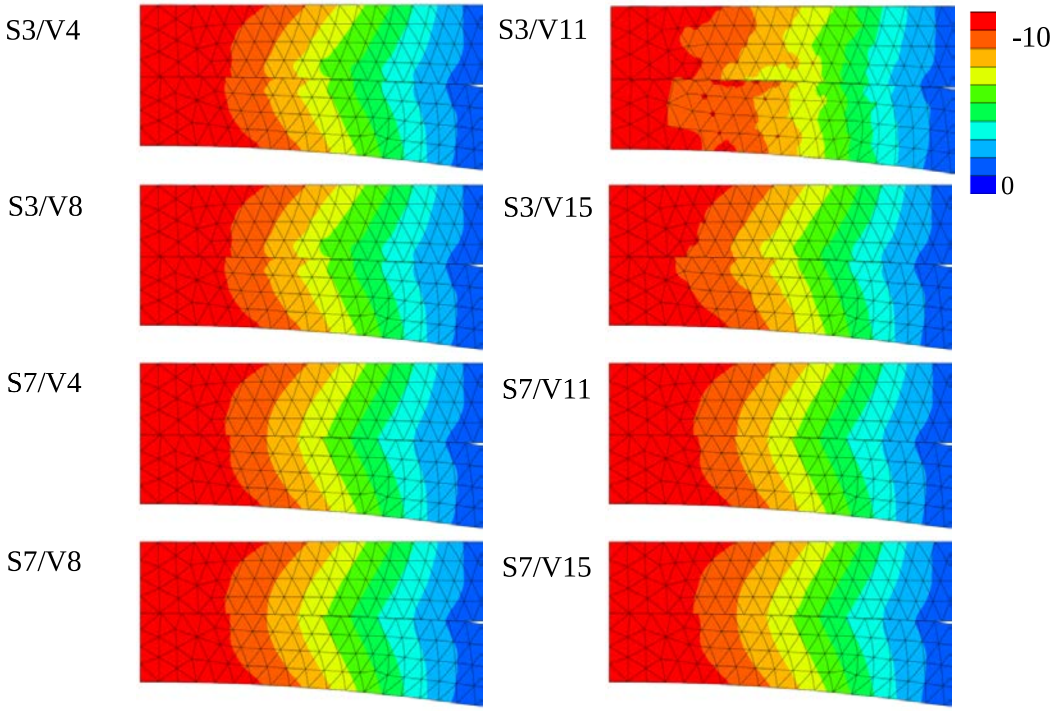


Fig. 4.5: Results for the plane strain contact problem. The isovalue fringe plots show 3rd principal stress for various combinations of surface and volume integration rules. Left column: results for TET10 element. Right column: results for the TET15 element. The first abbreviation in each label refers to the surface integration rule (S3, S7), while the second abbreviation refers to the volume integration rule (V4, V8, V11, V15). Overall, the S7 surface integration rule predicted more accurate stresses than the S3 rule. The volume integration rule did not have a significant effect on the predictions (with the combination S3/V11 a notable exception).

reason the remaining simulations were done with the S7/V8 rule for the TET10 element and the S7/V15 for the TET15 element.

4.4.2 Compression and sliding of a rigid cylinder on a deformable block

Overall reaction force on the cylinder provided an integrated measure of mesh convergence (Fig. 4.6). The quadratic tetrahedral element formulations demonstrated excellent agreement above 10,000 nodes with the HEX8 results. With the exception of the coarsest meshes, all values are within one percent of each other (Table 4.1, far right column). This is in stark contrast with the TET4 results, which did not appear converged even at the finest resolution and predicted much larger reaction forces.

Inspecting the reaction force as a function of horizontal displacement can give an indication of the sensitivity of the reaction force to the mesh distribution. At the lowest mesh resolution all formulations showed fluctuations in the reaction forces, however at the finest resolution the reaction force varied smoothly (Fig. 4.7). The difference between the minimum and maximum value was about 3% for each element formulation.

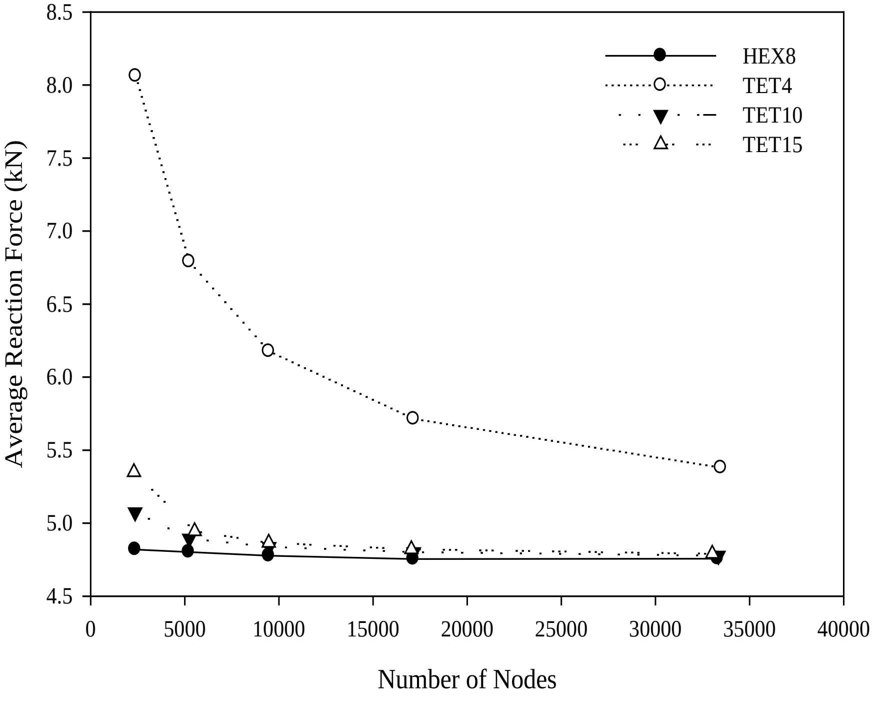


Fig. 4.6: Average reaction force on the rigid indenter while the indenter slides horizontally across the block, maintaining maximal vertical displacement. Average reaction forces are shown as a function of the number of nodes for a TET4 mesh, a TET10 mesh, a TET15 mesh and a HEX8 mesh. The TET10 and TET15 meshes above approximately 10,000 nodes are within 1% of the HEX results. The TET4 results predicted overall larger reaction forces and are not converged even at the finest mesh resolution.

Table 4.1: Results of the mesh convergence study for the contact sliding problem showing the number of nodes and elements, the number of time steps to complete the analysis, the number of matrix reformations (Refs) and right-hand-side (RHS) evaluations, and runtime in seconds, the average reaction force during horizontal displacement of the indenter, estimated contact area and peak contact pressure at midpoint of horizontal displacement.

HEX8	**Nodes**	**Elements**	**Steps**	**Refs.**	**RHS**	**Time (sec)**	**Force (kN)**	**Contact Area (mm^2)**	**Peak Contact Stress (MPa)**
Mesh1	2366	1800	40	79	995	53	4.82	1.6	-4.3
Mesh2	5208	4200	40	86	989	137	4.80	1.5	-4.7
Mesh3	9471	8000	40	84	1039	305	4.77	1.4	-4.7
Mesh4	17136	15000	40	84	1131	707	4.75	1.4	-4.8
Mesh5	33306	30000	40	88	1136	1931	4.76	1.4	-4.8
TET4									
Mesh1	2392	9524	40	109	1237	40	8.06	1.2	-6
Mesh2	5237	23259	40	84	1075	91	6.79	1.2	-7
Mesh3	9464	44049	40	82	1097	205	6.18	1.2	-6
Mesh4	17158	85561	40	87	1150	533	5.71	1.2	-7
Mesh5	33477	175945	41	98	1259	1785	5.38	1.2	-6
TET10									
Mesh1	2364	1128	40	93	1068	34	5.07	1.6	-4.7
Mesh2	5243	2824	40	81	1049	88	4.89	1.6	-5.2
Mesh3	9487	5521	40	82	1073	211	4.83	1.6	-5.0
Mesh4	17162	10331	40	90	1150	545	4.80	1.5	-5.4
Mesh5	33350	21049	42	99	1266	1748	4.78	1.5	-5.2
TET15									
Mesh1	2299	389	40	94	1155	27	5.35	1.7	-5.1
Mesh2	5521	999	40	87	1068	77	4.94	1.7	-5.2
Mesh3	9459	1815	40	83	1053	151	4.86	1.6	-5.2
Mesh4	17033	3383	40	83	1058	332	4.82	1.5	-5.3
Mesh5	33023	6694	40	84	1100	864	4.79	1.5	-5.2

The differences between the HEX8, TET10, and TET15 element formulations at a given displacement were even smaller and were less than 1%. Note again that the TET4 model predicted signifcantly larger reaction forces than the other three formulations.

Convergence behavior was comparable between the four element formulations. The number of stiffness matrix reformations and right-hand-side evaluations were similar. The runtimes for the HEX8 were similar, though consistently higher, than those for the TET4

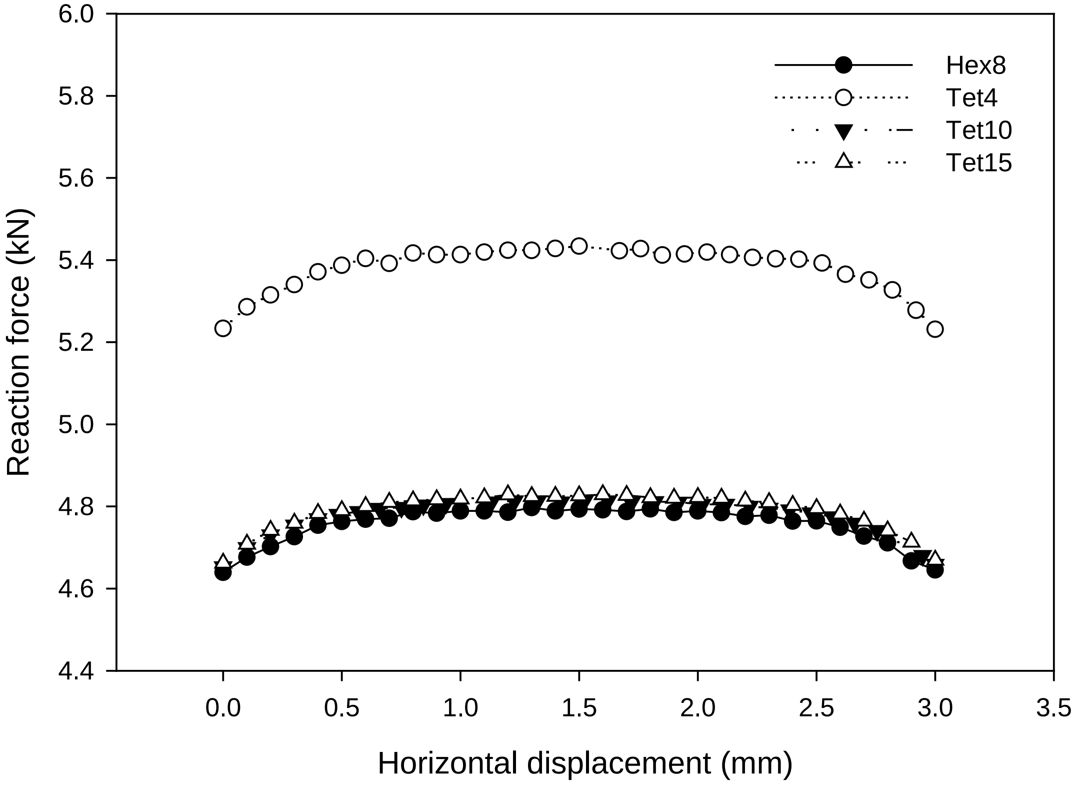


Fig. 4.7: Variation of the reaction force on the rigid indenter as a function of horizontal displacement for the finest mesh resolution of HEX8, TET4, TET10, and TET15 models. The variation is relatively small and smooth throughout the entire displacement and is reduced at the start and end by boundary effects. The HEX8, TET10, and TET15 predicted overall the same response, whereas the TET4 predicted much higher reaction forces due to the stiffer response that results from the spatially constant interpolation of strain.

and TET10 formulations. The TET15 models ran significantly faster. The predicted 3rd principal stress compared well between the quadratic tetrahedral and hexahedral models at the finest mesh resolutions (Fig. 4.8, right column). For the coarsest mesh resolutions, the TET15 and TET10 models were similar to the HEX8 results, though variations due to the irregularity of the mesh were noticeable (Fig. 4.8). The TET4 results were significantly higher and showed mesh dependent variations at all mesh resolutions.

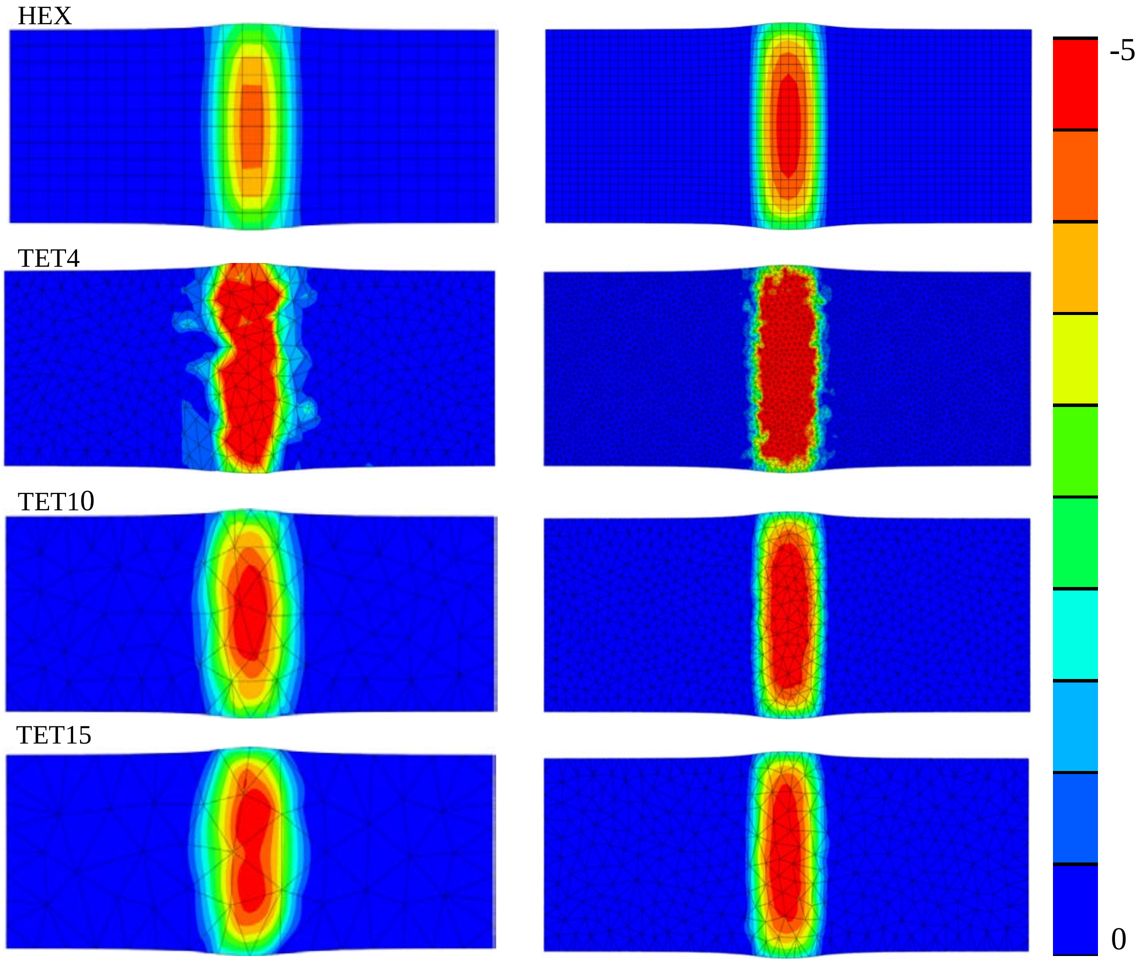


Fig. 4.8: 3rd principal stress fringe plots for two different mesh densities at the half-way point of horizontal displacement. From left to right: coarsest mesh, finest mesh density. From top to bottom: HEX8, TET4, TET10, and TET15 mesh. Stress predictions were very similar for HEX8, TET10, and TET15 element formulations at the finest level (right column). The TET4 results were overall higher and did not appear to be smooth even at the finest level of refinement.

4.4.3 Articular contact in the hip

Based on the results of the previous two models, the S7 surface integration rule was used to evaluate the contact integrals for both tetrahedral elements in this model. The V8 and V15 volume integration rules were used for the TET10 and TET15 elements, respectively. There was good qualitative agreement of the contact stress (measured by the 3rd principal stress) between the three models (Fig. 4.9). Quantitative comparisons confirmed this observation. For both the TET10 and TET15 models, the peak contact stress and the contact area on the acetabular cartilage compared well to the HEX8 results (Table 4.2).

In terms of performance, measured by the convergence statistics as well as overall runtime, all three models ran comparably, although the TET10 model did run slightly longer than both the HEX8 and TET15 models (Table 4.2).

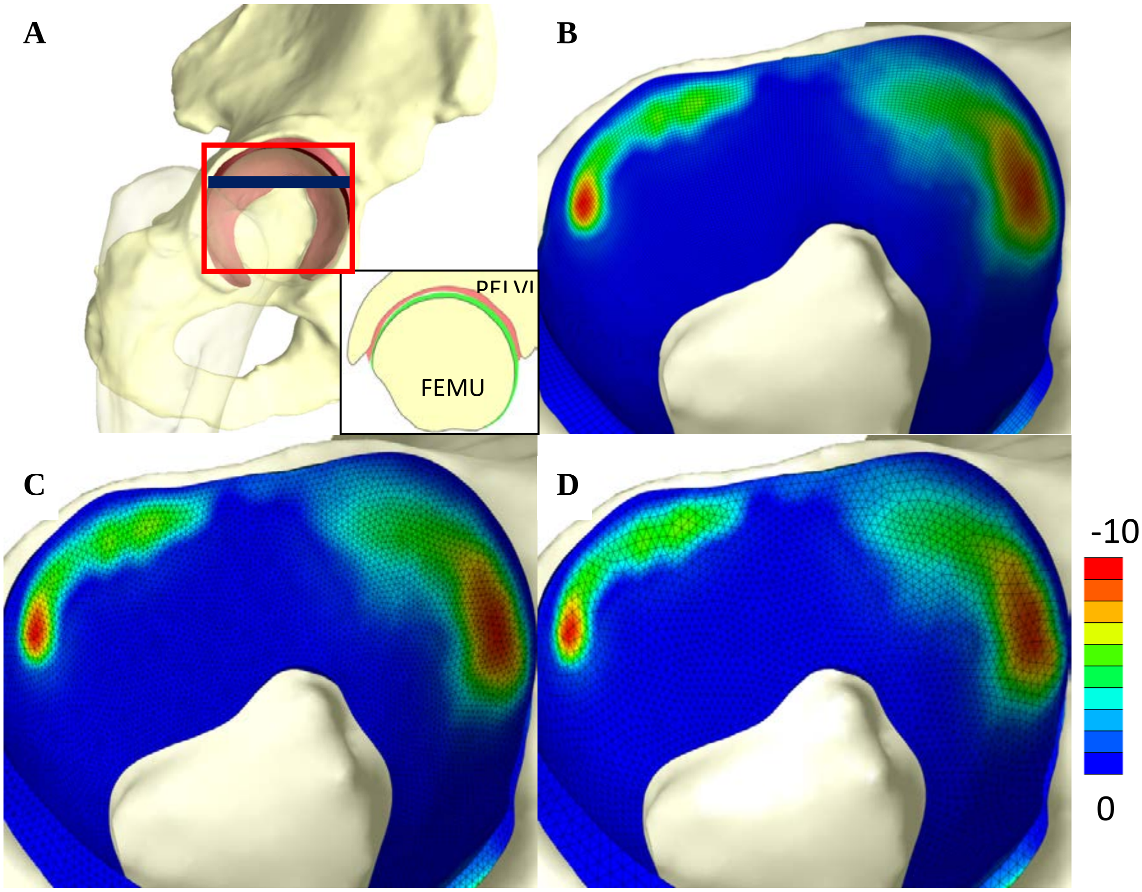


Fig. 4.9: A) Finite element model of the human hip, including the acetabulum, proximal femur and articular layers on the acetabulum and femoral head. Red square shows approximate area corresponding to contour plots. Blue line shows approximate location of inset, showing cross-section of model B-D) Fringe plots of 3rd principal stress for the acetabular cartilage discretized with B) HEX8 elements, C) TET10 elements, and D) TET15 elements. There was very good agreement between the 3rd principal stress and contact area predicted by the three models.

Table 4.2: Result of the hip model study where the results of a validated HEX8 model were compared with the predictions of a TET10 and TET15 model. The table lists the number of nodes and elements, peak contact stress, the contact area, the number of right-hand-side (RHS) evaluations, the number of stiffness reformations and the overall runtime (in minutes) for each of the three models.

	Nodes	**Elements**	**Peak Contact Stress (MPa)**	**Contact Area (mm^2)**	**RHS Evals**	**Reforms.**	**Run Time (min)**
HEX8	**144886**	**122148**	**-9.99**	**793**	**726**	**69**	**320**
TET10	**138081**	**83698**	**-9.93**	**792**	**892**	**80**	**374**
TET15	**139187**	**27905**	**-10.0**	**819**	**939**	**82**	**335**

4.5 Discussion

The results of this study demonstrate that quadratic tetrahedral elements are comparable both in computational cost and accuracy to the “gold standard” HEX8 elements for articular contact analysis. In some cases the TET15 models ran faster than the other models with comparable numbers of degrees of freedom, while in other cases they ran only slightly slower. It may seem surprising that these elements with more degrees of freedom are competitive in terms of computational cost, and in fact are faster than the TET4 model (e.g. the sliding contact example). Although the evaluation of the element quantities (i.e. element residual vector and stiffness matrix) is faster for low-order elements, the overall computational time may not follow the same pattern, as the examples clearly demonstrated. This can be explained by recognizing that many operations in the FE method (evaluation of global quantities, matrix assembly, etc.) are more proportional to the number of integration points than the number of degrees of freedom. This is especially true for contact problems, where the projection algorithm that determines the location of contact between surfaces can constitute a significant amount of the solution time. For a given number of nodes, a TET15 model will always have fewer elements than a similar TET10 or HEX8 and, although it uses more integration points than the TET10 or HEX8, the net sum often appears to be a reduction in the number of integration points. This may in part explain why the TET15 models ran faster in many cases.

The results of the plane strain contact analyses were very sensitive to the surface integration rule, and the S7 integration rule produced good results for the quadratic tetrahedrons. A possible explanation for this can be found by realizing that the accuracy of (Gaussian) integration rules is usually related to the ability to integrate polynomials exactly. For contact problems with non-conforming and possibly distorted surfaces, polynomials may not fit the surface tractions well and thus higher order integration rules are necessary to integrate the contact tractions accurately.

In contrast, the volume integration rule had little effect on the accuracy of the recovered stresses. All volume integration rules that were investigated are sufficient to obtain accurate results in linear analysis and it appears that, for the problems considered, the integration rules were also sufficiently accurate for nonlinear contact analysis. We did observe that the S3 surface integration rule in combination with the V11 volume integration rule for TET15 elements produced poor convergence as well as inaccurate results. We suspect that this combination does not lead to a stable element formulation. Despite the relatively small effect of the volume integration rule on the stress predictions, it may be prudent to consider the higher-order rules. The SPR recovery method, as with any least-squares method, requires a sufficient number of sample points to make the problem well-posed. In the present context this amounts to having a sufficient number of integration points surrounding each node. This may be challenging near boundaries, especially for articular contact problems, since the cartilage layers generally tend to be relatively thin. The result is that the V4 volume integration rule for TET10 prevents the use of a truly quadratic fit in the SPR recovery method. Although in this study we used a linear fit for the TET10 element and thus the V4 rule could have sufficed, there may be situations where the V8 rule is necessary to achieve accurate results. For the same reasons, for the TET15 elements, which contain additional monomials in their shape functions and thus require even more integration points that the TET10, it is prudent to consider the higher-order volume integration rule.

In this study we used “traditional” facet-on-facet based contact formulations (Laursen). In recent years, it has been demonstrated that mortar-based contact formulations have several advantages over traditional contact formulations (Puso, Laursen and Solberg). In essence, they minimize the error introduced by the non-conforming interface in either the displacements or the contact tractions. Yet, traditional formulations are widely used in many open-source and commercial finite element packages and thus evaluation of quadratic tetrahedral elements with traditional contact formulations has significant practical merit. These traditional formulations are known to give satisfactory results and, when used with an augmented Lagrangian method offer a good compromise between computational cost and accuracy. In addition, we anticipate that even for mortar-contact formulations the use of quadratic tetrahedral formulations will offer a distinctive advantage, foremost due to their ability to represent arbitrarily curved surfaces more accurately than linear triangular elements. Thus, we expect that the findings in this study remain valid, regardless of the contact formulation, though further investigations are necessary to confirm this.

In this study we only looked at the effects of frictionless contact. This is not seen as a strong limitation since friction is usually not considered in articular contact. That said, friction is a feature that is still under development in FEBio's contact formulations. Similarly, the effect of fluid motion in the tissue is of considerable interest in articular contact and was not considered here. The incorporation of fluid motion requires a biphasic analysis and the performance of quadratic tetrahedral formulations in this context is an important area for future investigation.

In the sliding contact case study we compared the quadratic tetrahedron results with linear tetrahedron results and demonstrated that, given the same number of nodes, the quadratic formulations, and especially the TET15 element, performed much better in all areas (i.e. accuracy, convergence, run-times) than the linear tetrahedron. In a way, this can be seen as yet another confirmation of the well-known mantra in finite element analysis that so-called *p*-refinement (i.e. increasing the polynomial order) is often a better refinement strategy than *h*-refinement (i.e. reducing the element size). Simply moving from a linear to a quadratic formulation resulted in vast improvements. In addition, the decreased run-times of higher-order tetrahedral formulations (for a given number of nodes), although perhaps surprising, is seen as an additional advantage of these quadratic tetrahedral formulations.

In conclusion, quadratic tetrahedral formulations offer an excellent alternative to HEX8 elements for simulation of articular contact. The TET15 element in particular appears to be very well suited for articular contact analysis.

4.6 Acknowledgments

Financial support from NIH grants #R01AR053344 and #R01GM083925 is gratefully acknowledged.

4.7 References

Abramowitz, M, and I.A. Stegun. *Handbook of Mathematical Functions*. 1964. Print.

Anderson, D. D., et al. "Physical Validation of a Patient-Specific Contact Finite Element Model of the Ankle." *J. Biomech.* 40.8 (2007): 1662-9.

Bergmann, G., et al. "Hip Contact Forces and Gait Patterns from Routine Activities." *J. Biomech.* 34.7 (2001): 859-71.

Bertrand, F.H., M.R. Gadbois, and P.A. Tanguy. "Tetrahedral Elements for Fluid Flow." *Intl. J. Num. Meth. Eng.* 33.6 (1992): 1251-67.

Bunbar, W.L., et al. "An Evaluation of Three-Dimensional Diarthrodial Joint Contact Using Penetration Data and the Finite Element Method." *J. Biomech. Eng.* 123.4 (2001): 333-40.

Cifuentes, A.O., and A. Kalbag. "A Performance Study of Tetrahedral and Hexahedral Elements in 3-D Finite Element Structural Analysis." *Finite Elem. Anal. Des.* 12.3-4 (1992): 313-18.

Danielson, Kent T. "Fifteen Node Tetrahedral Elements for Explicit Methods in Nonlinear Solid Dynamics." *Comput. Method. Appl. M.* 272 (2014): 160-80.

Das Neves Borges, P., et al. "Rapid, Automated Imaging of Mouse Articular Cartilage by Microct for Early Detection of Osteoarthritis and Finite Element Modelling of Joint Mechanics." *Osteoarthr. Cartilage* 22.10 (2014): 1419-28.

Delaunay, B. N. (1934) "Sur la Sphere" Vide. Izvestia Akademia Nauk SSSR, VII Seria, Otdelenie Matematicheskii i Estestvennyka Nauk Vol 7 pp.793-800. ""Sur La Sphere" Vide." *Izvestia Akademia Nauk SSSR, VII Seria, Otdelenie Matematicheskii i Estestvennyka Nauk* 7 (1934): 793-800.

Donzelli, P. S., et al. "Automated Adaptive Analysis of the Biphasic Equations for Soft Tissue Mechanics Using a Posteriori Error Indicators." *Intl. J. Num. Meth. Eng.* 34.3 (1992): 1015-33.

Dreischarf, M., et al. "Comparison of Eight Published Static Finite Element Models of the Intact Lumbar Spine: Predictive Power of Models Improves When Combined

Together." *J. Biomech.* 47 (2014): 1757-66.

Gee, M.W., et al. "A Uniform Nodal Strain Tetrahedron with Isochoric Stabilization." *Intl. J. Num. Meth. Eng.* 78 (2009): 429-43.

Hao, Z, et al. "The Effect of Boundary Condition on the Biomechanics of a Human Pelvic Joint under an Axial Compressive Load: A Three-Dimensional Finite Element Model." *J. Biomech. Eng.* 133.10 (2011).

Harris, M. D., et al. "Finite Element Prediction of Cartilage Contact Stresses in Normal Human Hips." *J. Orthop. Res.* 30.7 (2012): 1133-9.

Henak, C.L., et al. "Specimen-Specific Predictions of Contact Stress under Physiological Loading in the Human Hip: Validation and Sensitivity Studies." *Biomech. Model. Mechanobiol* 13.2 (2014): 387-400.

Hubsch, P. F., et al. "Adaptive Finite-Element Approach for Analysis of Bone/Prosthesis Interaction." *Med. Biol. Eng. Comput.* 33.1 (1995): 33-7.

Hughes, T.J.R. *The Finite Element Method*. Dover, 2000. Print.

Johnson, C. R., and R. S. MacLeod. "Adaptive Local Regularization Methods for the Inverse Ecg Problem." *Prog. Biophys. Mol. Biol.* 69.2-3 (1998): 405-23.

Johnson, Joshua E., et al. "Computationally Efficient Magnetic Resonance Imaging Based Surface Contact Modeling as a Tool to Evaluate Joint Injuries and Outcomes of Surgical Interventions Compared to Finite Element Modeling." *J. Biomech. Eng.* 136.4 (2014): 041002-02.

Keast, P. "Moderate-Degree Tetrahedral Quadrature Formulas." *Comput. Methods Appl. Mech. Eng.* 55.3 (1986): 339-48.

Kern, A. M., and D. D. Anderson. "Expedited Patient-Specific Assessment of Contact Stress Exposure in the Ankle Joint Following Definitive Articular Fracture Reduction." *J. Biomech.* (2015).

Khoshgoftar, M., et al. "The Sensitivity of Cartilage Contact Pressures in the Knee Joint to the Size and Shape of an Anatomically Shaped Meniscal Implant." *J. Biomech.* 48.8 (2015): 1427-35.

Laursen, T.A. *Computational Contact and Impact Mechanics*. Springer-Verlag, 2002. Print.

Laursen, T.A., and B.N. Maker. "Augmented Lagrangian Quasi-Newton Solver for Constrained Nonlinear Finite Element Applications." *International Journal for Numerical Methods in Engineering* 38 (1995): 3571-90.

Lo, S. H. "Volume Discretization into Tetrahedra-I. Verification and Orientation of Boundary Surfaces." *Comput. Struct.* 39.5 (1991): 493-500.

---. "Volume Discretization into Tetrahedra - Ii. 3d Triangulation by Advancing Front Approach." *Comput. Struct.* 39.5 (1991): 501-11.

Lohner, R. "Progress in Grid Generation Via the Advancing Front Technique." *Eng. Comput.* 12 (1996): 186-210.

Luczkiewicz, P., et al. "Influence of Meniscus Shape in the Cross Sectional Plane on the Knee Contact Mechanics." *J. Biomech.* 48.8 (2015): 1356-63.

Maas, S.A., et al. "Implementation and Verification of a Nodally-Integrated Tetrahedral Element in Febio." *SCI Technical Report*: University of Utah, 2011. Print.

Maas, S.A., et al. "Febio: Finite Elements for Biomechanics." *J. Biomed. Eng.* 134.1 (2012).

Maas, S.A., et al. "Febio User's Manual 1.5." 2012. Web.

Matthies, H., and G. Strang. "The Solution of Nonlinear Finite Element Equations." *Intl. J. Num. Meth. Eng.* 14 (1979): 1613-26.

McErlain, David D., et al. "Subchondral Cysts Create Increased Intra-Osseous Stress in Early Knee Oa: A Finite Element Analysis Using Simulated Lesions." *Bone* 48 (2011): 639-46.

Prakash, S., and C. R. Ethier. "Requirements for Mesh Resolution in 3d Computational Hemodynamics." *J. Biomech. Eng.* 123.2 (2001): 134-44.

Puso, M., and J. Solberg. "A Stabilized Nodally Integrated Tetrahedral." *International Journal for Numerical Methods in Engineering* 67 (2006): 841-67.

Puso, M.A., T.A. Laursen, and J. Solberg. "A Segment-to-Segment Mortar Contact Method for Quadratic Elements and Large Deformations." *Comput. Method. Appl. M.* 197.6-8 (2008): 555-66.

Shephard, M. S., and M. K. Georges. "Three-Dimensional Mesh Generation by Finite Octree Technique." *Intl. J. Num. Meth. Eng.* 32 (1991): 709-49.

Simo, J.C., and R.L. Taylor. "Quasi-Incompressible Finite Elasticity in Principal Stretches. Continuum Basis and Numerical Algorithms." *Computer methods in applied mechanics and engineering* 85.3 (1991): 273-310.

Tadepalli, S.C., A. Erdemir, and P.R. Cavanagh. "Comparison of Hexahedral and Tetrahedral Elements in Finite Element Analysis of the Foot and Footwear." *J.*

Biomech. 44 (2011): 2337-43.

Von Forell, GA, et al. "Low Back Pain: A Biomechanical Rationale Based on "Patterns" of Disc Degeneration." *Spine* 40.15 (2015): 1165-72.

Wan, C., Z Hoa, and S Wen. "The Effect of the Variation in Acl Constitutive Model on Joint Kinematics and Biomechanics under Different Loads: A Finite Element Study." *J. Biomech. Eng.* 135.4 (2013): 041002.

Weingarten, V.I. "The Controversy over Hex or Tet Meshing." *Mach. Des.* 66.8 (1994): 74-78.

Wrazidlo, W., et al. "An Alternative Method of Three-Dimensional Reconstruction from Two- Dimensional Ct and Mr Data Sets." *Eur. J. Radiol.* 12.1 (1991): 11-6.

Yang, T., and R. L. Spilker. "A Lagrange Multiplier Mixed Finite Element Formulation for Three-Dimensional Contact of Biphasic Tissues." *J. Biomech. Eng.* 129.3 (2006): 457-71.

Zienkiewicz, O.C., and J.Z. Zhu. "The Superconvergent Patch Recovery and a Posteriori Error Estimates. Part 1: The Recovery Technique." *Intl. J. Num. Meth. Eng.* 33 (1992): 1331-634.

CHAPTER 5

ADDITIONAL SIMULATIONS USING QUADRATIC TETRAHEDRAL ELEMENTS

5.1 Introduction

In the previous chapter, a study was presented that compared the performance and accuracy of the linear tetrahedral and two quadratic tetrahedral element formulations to a hexahedral element formulation in the context of large deformation contact analysis. The performance was measured in terms of overall runtime and convergence statistics. The accuracy was assessed by qualitative and quantitative comparison of the predicted contact stresses, which were recovered using the Superconvergent-Patch-Recovery (SPR) method. It was found that the quadratic tetrahedral formulations compared well, both in terms of performance and accuracy, and can be used as viable alternatives to the hexahedral formulation.

In this chapter, some additional results are presented that further corroborate these findings. First, for the cylinder-on-box sliding contact problem, a closer look at the reaction forces and stress predictions explored if the discretization of the rigid indenter had a discernible effect on the results. This was accomplished by comparing the results of the last chapter with results obtained from an analysis where the rigid indenter was represented by an implicitly defined surface. Next, the effect of using a structured versus an unstructured mesh for the deformable box was investigated. In the previous chapter, the hexahedral mesh for this problem was a structured mesh, whereas the tetrahedral meshes were unstructured. Even more, the mesh lines of the hexahedral meshes where aligned with the major axis of the cylindrical indenter. This may have given a significant advantage to the hexahedral mesh since the symmetry of the solution was recovered exactly for the hexahedral mesh, but not for the tetrahedral meshes that were used. Therefore some additional cases were explored where the tetrahedral meshes are structured and where the cylinder is not aligned with the mesh lines of the structured meshes. Lastly, an analysis of a Hertzian contact problem is presented. For large deformation contact problems, obtaining exact, analytical solutions is challenging. Therefore in the last chapter, the results were compared to the solution of a hexahedral model because this element is considered the de-facto standard in this area and is assumed to give reliable answers for a sufficiently refined mesh. However, when the problem is restricted to small deformations and linear elasticity, the predicted solutions can be compared with an analytical solution. This may give additional confidence that the mesh converged solutions are indeed the correct solutions for the element formulations considered here.

5.2 Methods

5.2.1 Implicit rigid indenter

In Chapter 4, a 20-node quadratic hexahedral mesh was used to represent the rigid indenter. The mesh resolution and the nodal coordinates of this mesh were chosen such that an approximately smooth surface was obtained. However, since the shape functions for the 20-node hex element are only quadratic, a perfectly smooth surface, where the shape function derivatives are continuous across element boundaries, cannot be obtained. Although mesh-refinement studies showed the effects on the results to be negligible, in this chapter, an additional analysis is presented where the indenter was represented by an implicit cylindrical surface. A new feature in FEBio was implemented that allows users to define a contact interface where one of the contacting surfaces is an implicitly defined rigid surface. The gap function and surface normal, which are required by the contact algorithm, can now be evaluated analytically. For a given node, the closest point on the implicit surface was calculated and the surface normal was evaluated at this point. A comparison study between the results obtained using the meshed indenter from Chapter 4 and the implicitly defined indenter described above investigated the effect of using the implicitly defined surface.

5.2.2 Structured meshing

As before, for all the additional cases presented here, the meshes are constructed such that approximately the same number of nodes were used for a given mesh resolution. For the analyses where the results from Chapter 4 are compared to new results, the unstructured tetrahedral meshes and the structured hexahedral meshes were taken from Chapter 4. For the cases where a structured tetrahedral mesh was required first, a structured hexahedral mesh was generated. Then, the structured tetrahedral meshes were constructed by splitting the hexahedral elements into six tetrahedrons. Aside from matching the number of nodes, the structured tetrahedral meshes also tried to match the number of nodes on the contacting surface to those on their structured counterpart. This was done since for contact problems, the surface mesh resolution is an important consideration and this way, a more direct comparison can be made between the structured and unstructured results (Table 5.1). All the additional mesh studies presented here used the implicit rigid indenter instead of a meshed indenter.

Table 5.1: The different mesh resolutions used in the comparison of structured and unstructured meshes. Only unstructured tetrahedral meshes were used in this study.

	unstructured mesh		structured mesh	
	nodes	surface nodes	nodes	surface nodes
HEX8	N/A	N/A	33306	1586
TET4	33477	3440	33670	3367
TET10	33350	3241	35035	3185
TET15	33023	2339	36227	2615

5.2.3 Hertzian contact

Analytical solutions for contact problems are available for some idealized geometries assuming infinitesimal strain conditions. Here, the case is considered where a rigid sphere of radius *R* indents a deformable half-space to depth *d*. The force on the rigid sphere to reach this depth is given by,

$$F = \frac{4E}{3\left(1-v^2\right)} R^{1/2} d^{3/2} . \tag{5.1}$$

where *E*, *v* are the Young's modulus, and Poisson's ratio of the deformable half-space, respectively. The contact area is a circle with radius,

$$a = \sqrt{Rd} . \tag{5.2}$$

The peak contact pressure occurs at the center of this circle and is given by

$$p_0 = \frac{3F}{2\pi Rd} . \tag{5.3}$$

To mimic the conditions of this idealized contact problem, a deformable cube of size *w* is indented by a rigid sphere of radius *R* at the center of the top surface of the cube. Due to the symmetry in the problem, only a quarter of the cube is modeled. This implies that the recovered contact force is only a quarter of the force obtained with (5.1). The size *w* must be chosen sufficiently large compared to the contact radius *a* in order to avoid the influence of the boundaries. At the same time, the element size *h* must be sufficiently small so that the contact area can be resolved accurately. The radius of the sphere is set to *R* = 10. FEBio does not have a linear elastic model, but the “isotropic elastic”, an implementation of the St-Venant Kirchhoff material, is a good approximation for small deformations. The material parameters were chosen to be *E* = 1, and *v* =0. The indentation depth was set to 0.01. The rigid sphere was represented by an implicit surface. Under these conditions, the exact reaction force and pressure can be calculated from equations (5.1) and (5.3), and yield *F/4* = 0.001054 and *p0* = 0.020132. The radius of contact can be evaluated using (5.2) and gives *a* = 0.316.

A sensitivity analysis was first conducted in order to determine suitable values for *w* and *h* using a rectangular, structured hexahedral mesh. As it turns out, the value of *w* must be chosen relatively large to obtain a good approximation to the analytical solution, so here a conservative measure of 10% deviation of the exact answers was deemed converged. From this analysis, a value of *w* = 3 was chosen for subsequent analyses. Next, structured and unstructured tetrahedral meshes were constructed and compared to the results obtained from the hexahedral mesh. Again, as before, the number of nodes was chosen as a metric for comparison. It was also attempted to match the number of surface nodes when comparing structured and unstructured tetrahedral meshes.

5.3 Results

5.3.1 Implicit rigid indenter

The use of the implicitly defined cylindrical indenter did not appear to affect the contact stress predictions, compared to the meshed cylinder that was used in the cylinder-on-box problem in Chapter 3 (Fig. 5.1). The contact stresses are both qualitatively and quantitatively in excellent agreement.

The total reaction force on the rigid indenter was also not affected. As was noted in Chapter 3 at the finest mesh resolutions, this force was not very sensitive to the mesh used (although it was much higher for the TET4 mesh, compared to the other element formulations). However, a measure that may be more sensitive to the mesh is the tangential component of the reaction force. Due to the model setup, this measure should be relatively small compared to the total reaction force, although it will not be identically zero due to

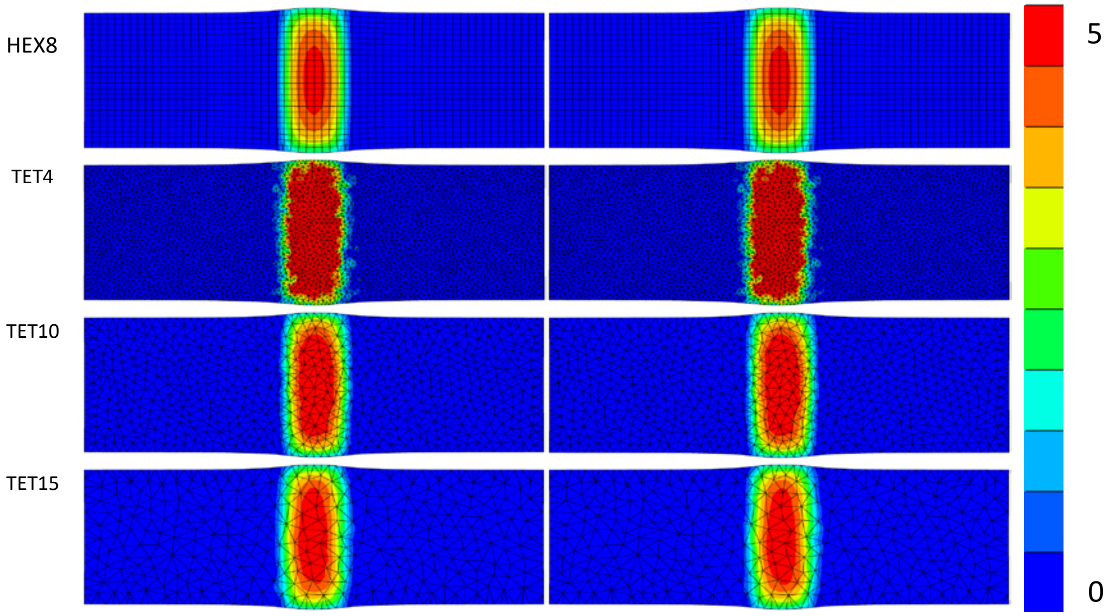


Fig. 5.1: Comparison of contact stress predictions for the highest mesh resolution between (left) results obtained from using a meshed indenter and (right) results obtained with the implicit indenter.

edge effects. Comparing the results between the meshed approach and the implicit approach at the finest mesh resolutions reveals that the tangential force was relatively unaffected (Fig. 5.2). The exception was the HEX8 result which did appear to show a small, yet systematic, discrepancy. Also, the tangential reaction force appeared to fluctuate the most for the HEX8 element, indicating that it may be sensitive to the mesh resolution, even at the finest mesh resolution considered in this study.

5.3.2 Structured tetrahedral meshes

The predicted contact stresses for the structured quadratic tetrahedral meshes compared well to those of the unstructured meshes, both qualitatively and quantitavely (Fig. 5.3). The results for the structured meshes do appear to recover the symmetry in the solution better than the results for the unstructured meshes. This is most likely due to the fact that the contact area cannot be represented exactly using an unstructured mesh. Aside

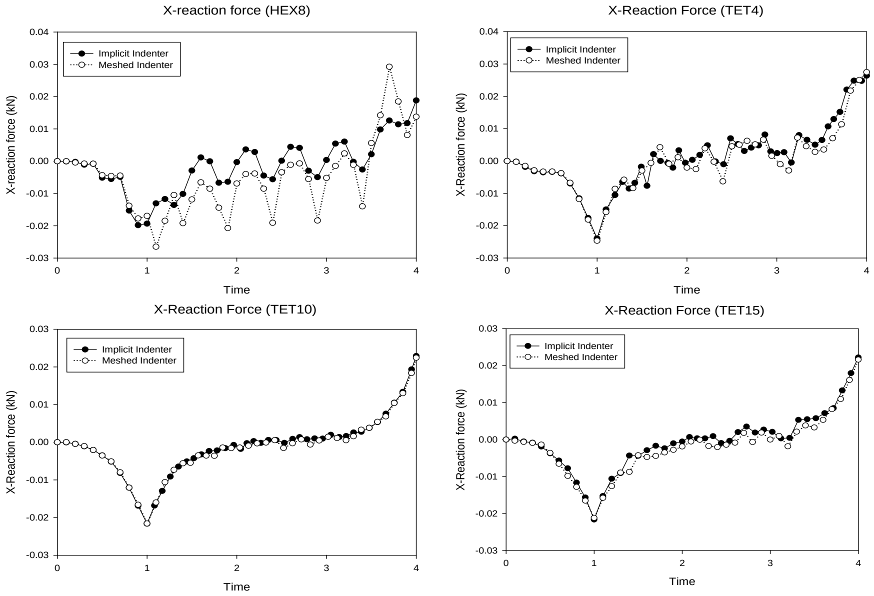


Fig. 5.2: Comparison of the tangential (X) component of the reaction force between the meshed indenter approach and the implicit indenter approach.

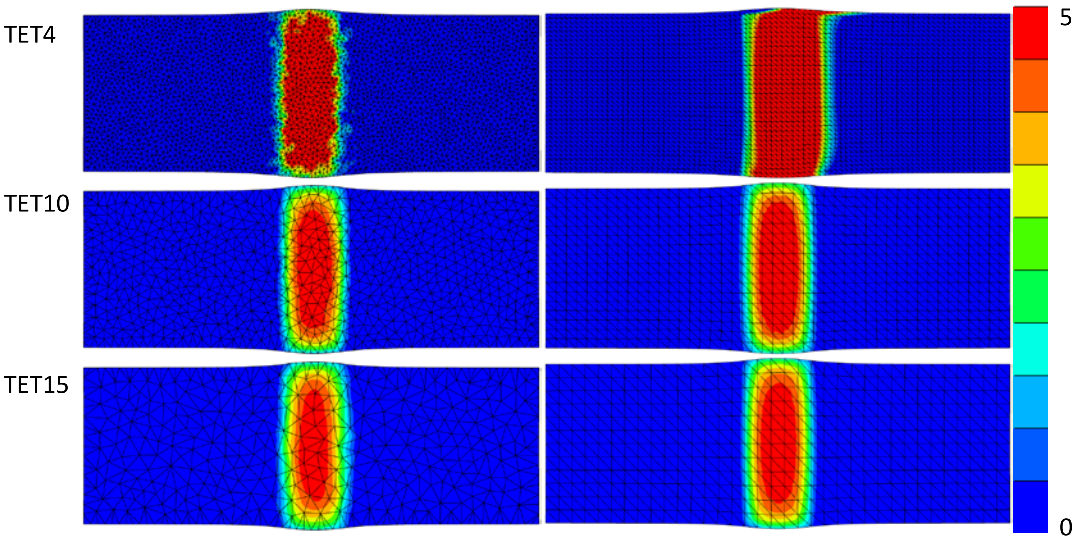


Fig. 5.3: Comparison of contact stresses between structured and unstructured meshes. (left) unstructured tetrahedral meshes, and (right) structured tetrahedral meshes.

from predicting a much higher contact stress, the linear tetrahedral mesh on the other hand failed to recover the symmetry in the solution near the outer edges of the mesh. It appears that the structured mesh strongly biases the result in these areas and is unable to recover an accurate solution.

5.3.3 Unaligned indenter

In the case where the indenter was angled with respect to the coordinate axes, the observations reported before remain valid. For both the structured and unstructured meshes, the quadratic results appear very similar to each other and to the hexahedral mesh result (Fig. 5.4). The linear tetrahedral mesh, on the other hand, predicted much higher contact stresses. Although the contact stresses did appear smoother for the structured linear tet mesh, it did have similar discrepancies near the edges as was observed in the previous analysis. As before, the contact stresses for the unstructured linear tet mesh are very irregular. This element formulation does not appear to be able to accurately capture the boundary of the contact area.

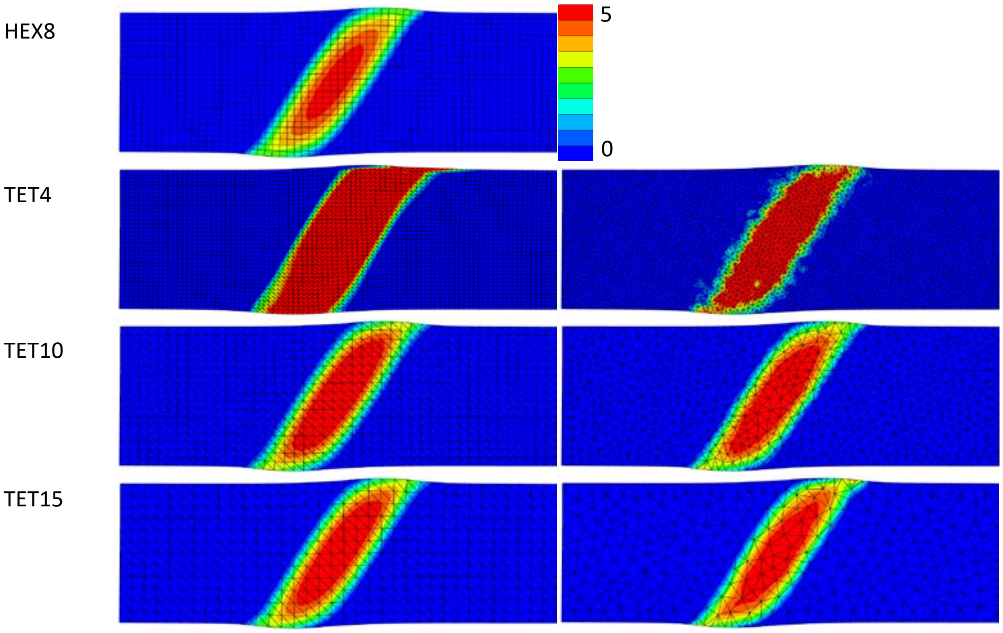


Fig. 5.4: Contact stresses for the unaligned indenter problem, where the cylindrical indenter makes a 30 degree angle with the vertical axis. The stresses are shown for a time point halfway of the horizontal displacement of the indenter.

5.3.4 Hertzian contact

The sensitivity analysis revealed that at a value of w = 3 (~ 10xa), the reaction force and pressure fell within 10% of the exact solution. For h = 0.1, F/4 = 0.00116 and p0 = 0.0201. Halving the value of h modified the solution only slightly. For h = 0.05, F/4 = 0.00114 and p0 = 0.0204. The latter case was taken as the reference solution to which the tetrahedral results were compared. At this resolution, the hexahedral mesh contained 226,981 nodes (3,721 surface nodes).

Qualitatively, all element formulations appear to provide good results for the contact stresses at the highest mesh resolution (Fig. 5.5). In all cases, the circular contact area was clearly discernable. The quadratic tetrahedral meshes did appear to represent the boundary of the contact area more accurately than the hexahedral mesh. Overall, the structured Tet15 mesh appeared to give the smoothest results. Past a mesh resolution of

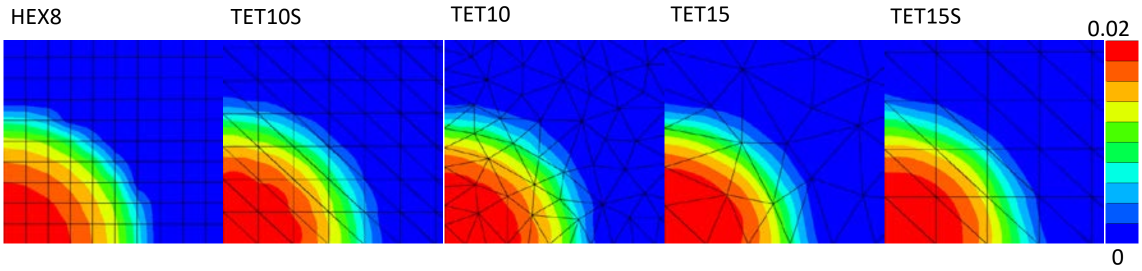


Fig. 5.5: Contact stresses near the area of indentation for various element formulations. The approximate area shown is 0.5x0.5 (the total surface area is 3x3).

about 50,000 nodes (corresponding to h = 0.08 for the hexahedral mesh), the predicted reaction force and peak contact pressure did not appear to change much, suggesting that the meshes are converged (Fig. 5.6). At the finest resolution (h = 0.05 for the hex mesh), the forces and peak contact pressure differed less than 1% and 4%, respectively, between the various element formulations.

5.4 Discussion

The additional analyses completed in this chapter further corroborate the findings of Chapter 4, namely that the quadratic tetrahedral elements perform quite well compared to the hexahedral element for large deformation contact problems and significantly better than the linear tetrahedral element. For the mesh converged solutions, the quadratic tetrahedral elements appear to give good predictions for contact stress and reaction force compared to the hexahedral mesh results.

Inspecting Fig. 5.2, it appears to show that the tangential component of the reaction force is smoother for the TET10 element than for the TET15 element, despite the fact that the TET15 element has more nodes per element. This can be explained by noting that, although both meshes had approximately the same number of nodes, the TET10 had a finer

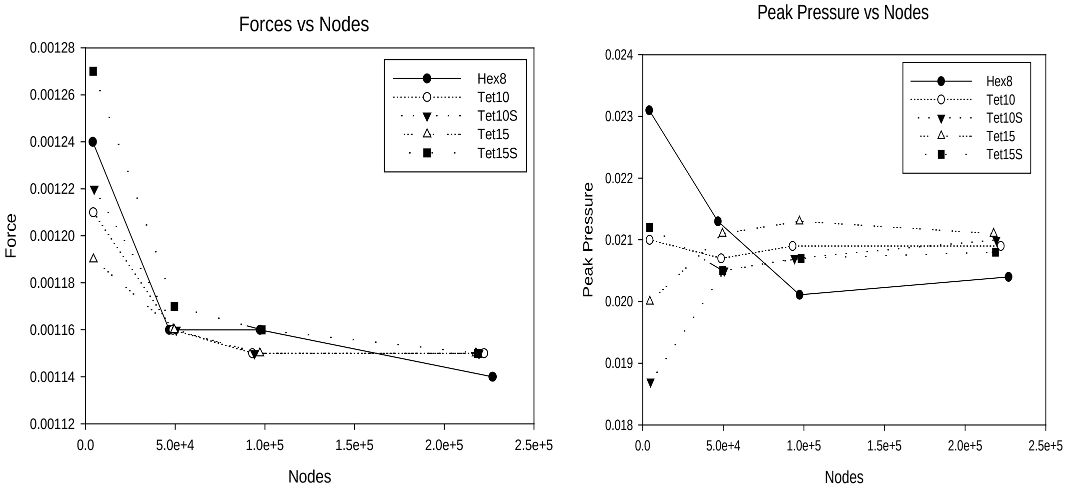


Fig. 5.6: Reaction force as a function of the total number of nodes in the mesh (Left). Peak contact pressure as a function of the total number of nodes. (Right)

surface mesh discretization than the TET15 mesh for a given number of nodes. Similarly, the HEX8 results appear to be the most sensitive to the mesh. Again this is most likely a result from the fact that the HEX8 had the coarsest mesh resolution for a given number of nodes. This result illustrates that the resolution of the contact surface may be an important factor to consider aside from the overall mesh resolution. This observation was the main motivation for trying to match the number of surface nodes between the structured and unstructured meshes, although the mesh topology did not always allow a close match.

Similarly for the Hertzian contact analysis, the number of nodes was matched for the four different mesh resolutions considered. Also, the number of surface nodes was matched between the corresponding meshes for the structured and unstructured cases. However, the TET15 meshes still had a lower surface mesh resolution than the TET10 meshes (Table 5.2). Despite this disadvantage, the TET15 results appeared to be comparable to the TET10 results. In fact, the qualitative comparison of the contact stress (Fig. 5.5) shows that the TET15 results appeared to be slightly better than the TET10

Table 5.2: Mesh resolutions and results for the mesh convergence study of the Hertzian contact analysis problem.

		nodes	elems	surf nodes	F/4	p0	runtime (sec)
TET10 - US	mesh1	4347	2383	437	0.00121	0.021	1
	mesh2	48670	32300	2153	0.00116	0.0207	35
	mesh3	93025	63482	3285	0.00115	0.0209	114
	mesh4	222166	155444	6197	0.00115	0.0209	670
TET10 - S	mesh1	4851	3000	441	0.00122	0.0187	2
	mesh2	50625	34848	2025	0.00116	0.0205	39
	mesh3	94221	65856	3249	0.00115	0.0207	105
	mesh4	219373	155952	5929	0.00115	0.021	480
TET15 - US	mesh1	4407	845	279	0.00119	0.02	1
	mesh2	49461	10334	1683	0.00116	0.0211	22
	mesh3	97217	20893	2433	0.00115	0.0213	64
	mesh4	217671	47693	4095	0.00115	0.0211	283
TET15 - S	mesh1	4281	864	241	0.00127	0.0212	1
	mesh2	49551	10752	1601	0.00117	0.0205	22
	mesh3	98259	21600	2481	0.00116	0.0207	62
	mesh4	218841	48672	4161	0.00115	0.0208	250

results. Even more, the overall runtime for the TET15 was considerably lower than the TET10 and HEX results for the same number of nodes (Table 5.2). These observations further strengthen the conclusion from Chapter 4 that the TET15 may be a better element to consider than the TET10 element for large deformation contact analysis.

For the Hertzian contact analysis, it must also be noted that the resolution of the contact area was relatively small, even for the highest mesh resolution considered. This was a consequence of the fact that the overall size of the box had to be sufficiently large in order to avoid boundary effects that would have caused deviations from the analytical solution. A further refinement of the mesh in the contact area would most likely improve the results even more. In order to reduce the computational time, this problem would greatly benefit from an inhomogeneous mesh that is locally refined near the area of contact. In practice, this would be the preferred approach, although in these studies, homogenous meshes were considered to simplify the analyses.

CHAPTER 6[1]

A GENERAL FRAMEWORK FOR APPLICATION OF PRESTRAIN TO COMPUTATIONAL MODELS OF BIOLOGICAL MATERIALS

6.1 Abstract

It is often important to include prestress in computational models of biological tissues. The prestress can represent residual stresses (i.e. stresses that exist after the tissue is excised from the body) or *in situ* stresses (i.e. stresses that exist *in vivo*, in the absence of loading). A prestressed reference configuration may also be needed when modeling the reference geometry of biological tissues *in vivo*. This research developed a general framework for representing prestress in finite element models of biological materials. It is assumed that the material is elastic, allowing the prestress to be represented via a prestrain. For prestrain fields that are not compatible with the reference geometry, the computational framework provides an iterative algorithm for updating the prestrain until equilibrium is satisfied. The iterative framework allows for enforcement of two different constraints: elimination of distortion in order to address the incompatibility issue, and enforcing a specified *in situ* fiber strain field while allowing for distortion. The framework was

[1] Reprinted from *Journal of the Mechanical Behavior of Biomedical Materials*, Volume 61, Pages 499-510, Maas SA, Erdermir A, Halloran JP, Weiss JA, “A General Framework for Application of Prestrain to Computational Models of Biological Materials”, with permission from ELSEVIER.

implemented as a plugin in FEBio (www.febio.org), making it easy maintain the software and to extend the framework if needed. Several examples illustrate the application and effectiveness of the approach, including the application of *in situ* strains to ligaments in the Open Knee model (simtk.org/home/openknee). A novel method for recovering the stress-free configuration from the prestrain deformation gradient is also presented. This general purpose theoretical and computational framework for applying prestrain will allow analysts to overcome the challenges in modeling this important aspect of biological tissue mechanics.

6.2. Introduction

Experimental observations show that connective tissues such as ligaments, tendons and skeletal muscle retract when excised from the body. This retraction is due to *in situ strain* -strain that exists *in vivo* in the absence of loading in the *reference configuration*. The strain and associated stress is relieved when the tissue is removed from the body, yielding a relatively *stress-free configuration*. *In situ* strains for ligaments in diarthrodial joints are in the range of 3-10% (Gardiner, Weiss and Rosenber; Woo et al.) and it has been shown that they contribute to the stability of joints (Ellis et al.; Lujan, Dalton, et al.). *Residual strain* (i.e. strain that exists in the tissue after it is excised from the body) is observed in many tissues such as arteries (Chuong and Fung), mitral leaflets (Rausch and Kuhl) and myocardium (Wang et al.; Omens and Fung; Guccione, McCulloch and Waldman). Often, geometries of biologically tissues are acquired *in vivo* and consequently the reference configuration cannot be assumed stress-free. All these various forms of prestrain, i.e. strain that exists in the reference configuration of the body of interest, can contribute significantly to the mechanical response of the system. Inclusion is often necessary in computational models of biological tissues to obtain reasonable predictions of tissue mechanics.

One class of previously reported methods for accommodating prestrain in finite element modeling can be described as deforming a stress-free configuration to induce stress in a desired/known reference configuration. For example, in Balzani, Schroder and Gross this is accomplished by closing the geometry representing a radially cut artery, using a special type of spring element. In Rausch and Kuhl, a stress-free configuration of mitral leaflets is stretched to conform to the *in vivo* reference configuration. In this case, the deformation map from stress-free to prestressed reference configuration was assumed to be known. This class of methods requires a sequence of forward analyses and can be executed with any finite element analysis software. The drawback is that they rely on the existence and knowledge of a stress-free configuration. In practice however, obtaining a stress-free configuration can be challenging, and there is no guarantee that it exists. For instance, in the case of arteries, the opening angle experiment was once believed to relieve the residual stress (Fung and Liu). However, it was later shown that a single cut does not relieve all the residual stress (Vossoughi, Hedzaji and Borris; Greenwald et al.).

In another class of methods, prestress is accounted for directly in the reference configuration without the requirement that a stress-free state exists or is known *a priori*. These methods are especially useful in the context of patient- and subject-specific modeling, since in these cases finite element models are often constructed based on image data acquired in the reference configuration. Usually the materials are assumed to be elastic and the prestress is then defined via a prestrain. The methods in this class primarily differ from each other in the definition of the prestrain. In Alastrue et al., the deformation gradient is taken from an analytical solution for the bending of a cylinder. In Weiss et al. and Dhaher, Kwon and Barry it was obtained indirectly from experimental data. In Gee, Forster and Wall and later in Weisbecker, Pierce and Holzapfel and Grytz and Downs, the prestrain deformation gradient was obtained by solving an inverse finite element problem: given the *in vivo* reference configuration and the *in vivo* loads, find the deformation gradient that generates the stresses in the reference configuration required to balance the applied loads. This approach requires an iterative solution due to the nonlinearity of large deformation elasticity. The method by Bols et al. is also similar in that regard, except that it attempts to recover the stress-free configuration. In general, biological tissues can have residual stresses in addition to prestress, and the methods mentioned above can be used to accommodate all forms of prestrain in a single analysis. For instance in Pierce et al. the general prestressing algorithm by Weisbecker, Pierce and Holzapfel is used to account for both the residual stress and the prestress in the reference configuration of an artery. A shortcoming of the methods that start from a prestrain deformation gradient is that it cannot always be guaranteed that the induced prestress is in equilibrium with the given reference configuration. Thus, it must be verified that the applied prestrain results in equilibrated stresses. Ideally, a method to compensate for any incompatibility should be available.

The objective of this study was to develop and implement a general purpose computational framework for modeling prestrain in finite element models of biological tissues. The framework uses a prestrain gradient approach that does not require the knowledge or computability of a stress-free reference configuration. The manuscript details the theoretical foundation as well as the computational aspects of the framework. We demonstrate that some of the previously reported methods for applying prestrain can be recovered as special cases of the framework. We also describe a method for recovering prestrain from sparse experimental data, and a method for recovering the global stress-free state from its deformation gradient. Examples illustrate the application of the framework to several test problems, including the application of prestrain to a finite element model of the knee from the Open Knee project (Erdemir and Sibole; Erdemir; Erdemir). The framework was implemented in the freely available finite element software FEBio (Maas et al.) and can be used with any of the elastic constitutive models.

6.3. Methods

6.3.1 Theoretical background

Consider a body in its prestressed reference configuration that is subjected to applied loading and subsequently deforms into a loaded configuration (Fig. 6.1). The deformation map $\varphi(\mathbf{X})$, which maps the material coordinates $\mathbf{X}$ to the corresponding

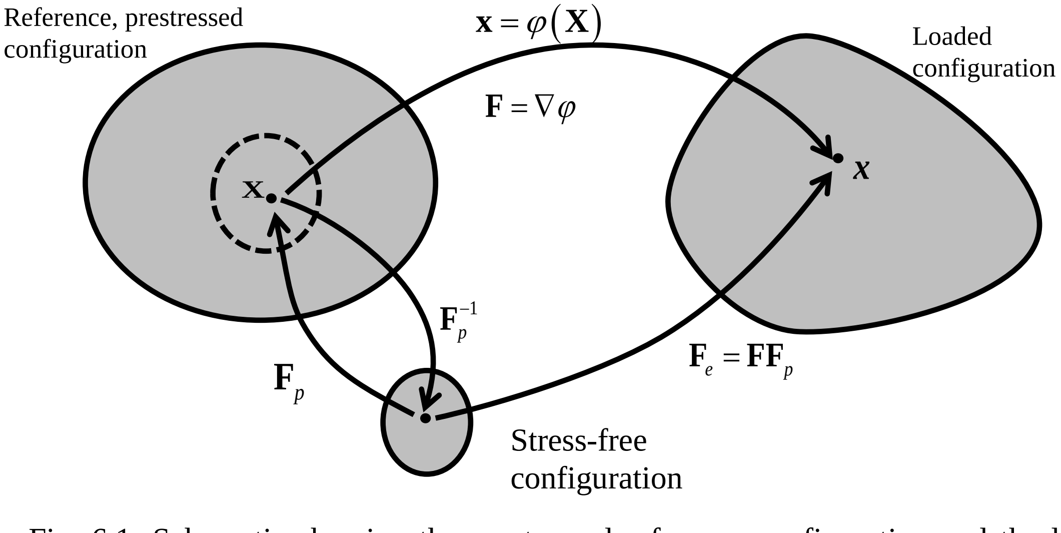

Reference, prestressed configuration
$\mathbf{x} = \varphi(\mathbf{X})$
Loaded configuration
$\mathbf{F} = \nabla\varphi$
X
x
$\mathbf{F}_p^{-1}$
$\mathbf{F}_p$
$\mathbf{F}_e = \mathbf{F}\mathbf{F}_p$
Stress-free configuration

Fig. 6.1: Schematic showing the prestressed reference configuration and the loaded, deformed configuration. At each point in the reference configuration, it is assumed that a small (potentially infinitesimal) neighborhood can be found for which a local deformation map exists that takes the neighborhood to a stress-free state.

spatial coordinates $\mathbf{x} = \boldsymbol{\varphi}(\mathbf{X})$, has a deformation gradient $\mathbf{F} = \nabla_0 \boldsymbol{\varphi}$. It is assumed that for each material point, a local stress-free virtual configuration can be found in the sense of (Johnson and Hoger). The gradient of the local mapping from the stress-free to the prestressed reference configuration is represented by $\mathbf{F}_p$, which will be referred to as the *prestrain gradient.*

The total elastic deformation gradient $\mathbf{F}_e$ is determined by the composited deformation gradient,

$$\mathbf{F}_e = \mathbf{F}\mathbf{F}_p \; . \tag{6.1}$$

It is further assumed that the constitutive response of the material is hyperelastic. As shown by Johnson and Hoger, in this case the Cauchy stress $\boldsymbol{\sigma}$and spatial elasticity tensor $\boldsymbol{c}$ can now be evaluated using $\mathbf{F}_e$:

$$\boldsymbol{\sigma} = \mathscr{F}(\mathbf{F}_e), \qquad \boldsymbol{c} = \mathscr{T}(\mathbf{F}_e) \; , \tag{6.2}$$

where $\mathscr{F}$ and $\mathscr{T}$ denote the “natural” material response functions (i.e. the response functions obtained from deforming the material from a stress-free reference configuration). For a finite element software such as FEBio that evaluates these quantities in the spatial frame, accommodating prestrain via the prestrain gradient approach is relatively straightforward.

In general, the inverse deformation gradient $\mathbf{F}_p^{-1}$ that renders the neighborhood of a point in the reference configuration stress-free can only be defined locally. Thus $\mathbf{F}_p$

cannot be obtained from the derivative of a mapping from a stress-free configuration to the prestressed configuration. This is a strong motivation for using the prestrain gradient approach in this work. Of course, if such a mapping is available, its gradient can be computed directly and then used to evaluate the material response via Eq. (6.2).

Finally, the Cauchy stress in the prestressed configuration $\boldsymbol{\sigma}_p = \mathscr{F}\left(\mathbf{F}_p\right)$ must satisfy equilibrium in the absence of any subsequent deformation (i.e. $\mathbf{F} = \mathbf{1}$) . This implies that, in the absence of body forces, $\boldsymbol{\sigma}_p$ must satisfy the equations representing conservation of linear and angular momentum in the interior of the prestressed domain:

$$\begin{aligned} &\operatorname{div}\left(\boldsymbol{\sigma}_p\right) = \mathbf{0}, \\ &\boldsymbol{\sigma}_p = \boldsymbol{\sigma}_p^T. \end{aligned} \tag{6.3}$$

In addition, the free boundaries must be traction free:

$$\boldsymbol{\sigma}_p \cdot \mathbf{n} = \mathbf{0} \; . \tag{6.4}$$

Here, $\mathbf{n}$ is the unit outward normal vector at the boundary. When the stress induced by the prestrain gradient satisfies Eqs. (6.3) and (6.4) the prestrain gradient is considered *compatible* with the reference geometry.

6.3.2 The prestrain gradient approach

The proposed framework starts from knowledge of an initial target for the prestrain gradient, an initial reference configuration, and the boundary conditions and loads that exist in this configuration. As noted above, this is the most convenient starting point when the prestrain gradient is known (e.g. from a forward FE analysis or from an analytical solution) or can be obtained from experiment. At the same time, experimental measurement of the complete prestrain gradient can be difficult, so the full prestrain gradient may not always be available. For instance, when measuring ligament strain, often the fiber stretch is measured (Weiss et al.; Lujan, Lake, et al.; Lujan, Dalton, et al.; Gardiner, Weiss and Rosenber; Gardiner and Weiss; Ellis et al.; Dhaher, Kwon and Barry). In this case, additional modeling assumptions are necessary to fill in the missing information. Similarly, assumptions are often made about the form of the prestrain gradient when measuring strain in arteries (Balzani, Schroder and Gross; Balzani, Schroder and Gross; Holzapfel and Ogden; Labrosse et al.; Raghavan et al.) and the heart (Wang et al.; Omens, McCulloch and Criscione; Taber et al.; Taber and Chabert). The result of these assumptions is an approximate prestrain field that may not be compatible with the initial reference configuration. In a finite element analysis, the application of an incompatible prestrain field results in a distortion of the reference geometry which also alters the effectively applied prestrain gradient. It is usually not desirable to have both a distortion of the reference configuration and an effective prestrain gradient that differs from the applied prestrain gradient as this may complicate the interpretation of the results. Depending on the specific goal and the input data, the analyst may wish to eliminate distortion and thus needs to find an alternative prestrain gradient field that is compatible with the original reference configuration. Alternatively, the goal may be to enforce the given prestrain gradient field as closely as possible and in this case a new reference configuration needs to be found that is compatible with the applied prestrain gradient field.

For both goals, the framework uses an iterative mechanism that updates the prestrain gradient field in order to accomplish the desired goal. In the following sections the deformation gradient that corresponds to the distortion of the geometry is denoted by **F**. Our computational framework assumes that the user provides an initial target for the prestrain gradient, notated as $\hat{\mathbf{F}}_p$. Then the prestrain gradient $\mathbf{F}_p$ that results in an effective prestrain gradient $\mathbf{F}_{p,eff}$ is:

$$\mathbf{F}_{p,eff} = \mathbf{F}\mathbf{F}_p, \tag{6.5}$$

such that $\mathbf{F}_{p,eff}$ is compatible with the possibly distorted reference configuration and is calculated using an iterative algorithm (Fig. 6.2). Users can provide either the full tensor $\hat{\mathbf{F}}_p$ or provide a simplified functional form that calculates $\hat{\mathbf{F}}_p$ from given data. In either case, $\hat{\mathbf{F}}_p$ is held constant during the iterative loop. A forward finite element analysis is then executed which results in a distortion of the mesh with deformation gradient $\mathbf{F}^k$. After the forward analysis, $\mathbf{F}_p$ is updated. Since the update depends on the desired goal, the user selects one of the by default provided update rules (or implements a new one). After $\mathbf{F}_p$ is updated, a new forward analysis is performed. This iterative procedure continues until $\mathbf{F}_p$ converges to a stable value. In our current implementation, the convergence criterion is

$$\left\| \mathbf{F}_p^{k+1} - \mathbf{F}_p^k \right\| < \varepsilon . \tag{6.6}$$

After the algorithm converges, the initial reference geometry is replaced with the distorted geometry, $\mathbf{F}_p \leftarrow \mathbf{F}\mathbf{F}_p$. $\mathbf{F}$, which measures the deformation of the final distortion, is reset to the identity tensor. Now $\mathbf{F}_p$ is compatible with the new reference geometry. This

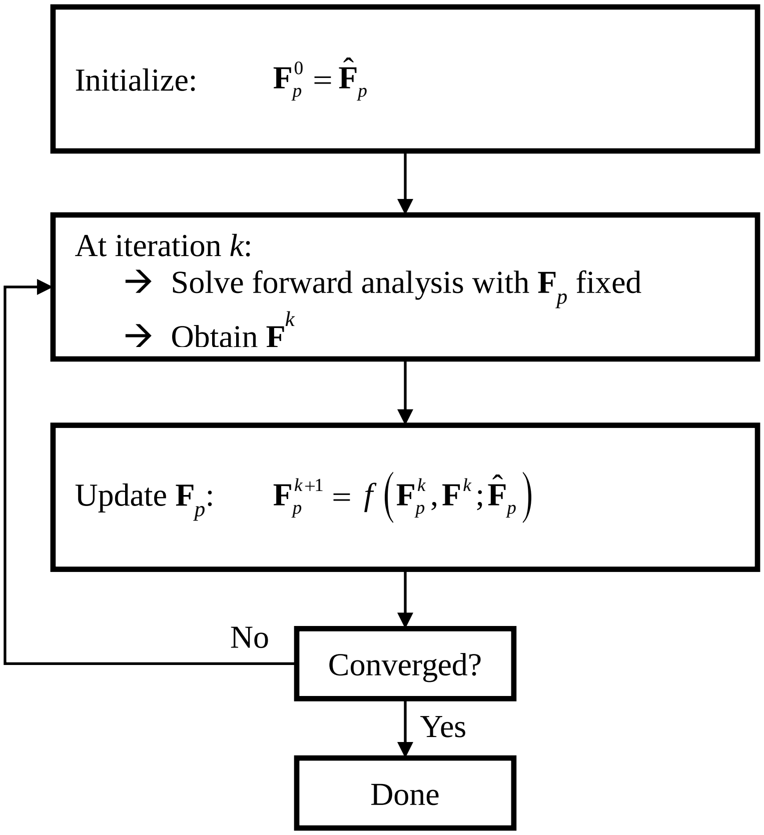

Initialize: $\mathbf{F}_p^0 = \hat{\mathbf{F}}_p$
At iteration k:
→ Solve forward analysis with $\mathbf{F}_p$ fixed
→ Obtain $\mathbf{F}^k$
Update $\mathbf{F}_p$: $\mathbf{F}_p^{k+1} = f\left(\mathbf{F}_p^k, \mathbf{F}^k; \hat{\mathbf{F}}_p\right)$
No
Converged?
Yes
Done

Fig. 6.2: Flowchart of numerical algorithm, showing how the prestrain gradient is found iteratively by solving a series of forward FE problems. In the forward FE model, the prestrain gradient is held constant. After each forward analysis, the prestrain gradient is updated.

step is necessary to ensure that any subsequent loading is applied to the compatible reference geometry and not the incompatible initial reference geometry.

This approach offers a very flexible approach to applying a prestrain gradient field and, as illustrated below, can be used to implement different algorithms for applying prestrain in a finite element model. The main difference between these algorithms will lie in two aspects: how $\hat{\mathbf{F}}_p$ is constructed from given data and how $\mathbf{F}_p$ is updated. In Sections 6.3.2.1 and 6.3.2.2, two particular update rules will be considered as examples that illustrate the possible applications of the proposed framework. In Section 6.2.2.3 a form for $\hat{\mathbf{F}}_p$ written in terms of fiber stretches will be presented, and that form will be used in the examples in Section 6.4.

6.3.2.1. Elimination of distortion

In this application the goal is to eliminate the distortion induced by the incompatibility of the initial prestrain gradient. As explained in Section 6.2.1, the iterative algorithm alternates between a forward FE analysis (with $\mathbf{F}_p$ fixed) and an update to $\mathbf{F}_p$. At iteration *k*, the forward analysis induces a distortion of the reference geometry, characterized by a deformation gradient $\mathbf{F}^k$. This is approximately equivalent to distorting the initial reference configuration backwards via $\left(\mathbf{F}^k\right)^{-1}$, such that the distorted geometry coincides with the given geometry. From this altered initial reference configuration, the effective prestrain gradient is given by $\mathbf{F}^k_{p,eff} = \mathbf{F}^k \mathbf{F}^k_p$. The update rule then follows by requiring that the new prestrain gradient coincides with the effective prestrain gradient, i.e. $\mathbf{F}^{k+1}_p = \mathbf{F}^k_{p,eff}$. From this, the update rule follows directly:

$$\mathbf{F}^{k+1}_p = \mathbf{F}^k \mathbf{F}^k_p \ . \tag{6.7}$$

A new forward analysis is then executed. The process is repeated until $\mathbf{F}_p$ converges. The result of this approach is a prestrain gradient field that is compatible with the original reference configuration, thus eliminating distortion.

Note that the $\mathbf{F}_p$ in general will not be identical to $\hat{\mathbf{F}}_p$. This is the price that must be paid for a prestrain field that is compatible with the original reference configuration. An obvious question is then is it possible to find a reference configuration that allows initial prestrain to be enforced exactly. This question motivates the next application.

6.3.2.2. Enforcement of a prestrain gradient field

Enforcing the initial prestrain gradient $\hat{\mathbf{F}}_p$ is equivalent to stating that a reference configuration is to be found in which the effective prestrain gradient is equal to the initial prestrain gradient, i.e.

$$\mathbf{F}_{p,eff} = \hat{\mathbf{F}}_p \,. \tag{6.8}$$

At iteration *k* of the algorithm, the mesh will distort with deformation gradient $\mathbf{F}^k$ and the effective prestrain will be $\mathbf{F}^k_{p,eff} = \mathbf{F}^k \mathbf{F}^k_p$. Keeping $\mathbf{F}^k$ fixed during the update phase, Eq. (6.8) can be enforced by choosing $\mathbf{F}^{k+1}_{p,eff} = \mathbf{F}^k \mathbf{F}^{k+1}_p = \hat{\mathbf{F}}_p$. Solving this equation for $\mathbf{F}^{k+1}_p$ yields the following update rule:

$$\mathbf{F}^{k+1}_p = \left(\mathbf{F}^k\right)^{-1} \hat{\mathbf{F}}_p \,. \tag{6.9}$$

In general, exact enforcement of a particular $\hat{\mathbf{F}}_p$ is an ill-posed problem. Small variations in $\hat{\mathbf{F}}_p$ can lead to large distortions of the mesh, or in the worst case, may prevent the algorithm from converging. This is expected as an arbitrary prestrain gradient cannot be enforced on an arbitrary reference geometry. Progress can be made by appropriate regularization. For instance, instead of enforcing the full initial prestrain gradient, only part of the tensor is enforced. This compromise allows some distortion, but still enforces part of the initial prestrain gradient that is most important for reproducing e.g. experimental measurements. In the examples below it is assumed that the local fiber stretch is known (e.g. from experimental data). Thus, an update rule that only enforces the given fiber stretches is derived next.

The fiber stretch $\hat{\lambda}_p$ induced by the initial prestrain gradient $\hat{\mathbf{F}}_p$ is given by

$$\hat{\lambda}_p \hat{\mathbf{a}}_p = \hat{\mathbf{F}}_p \cdot \mathbf{a}_0 . \tag{6.10}$$

Here, $\mathbf{a}_0$ is the fiber vector in the stress-free configuration and is in general unknown. Since the fiber vector $\hat{\mathbf{a}}_p$ in the initial reference configuration is usually known, we can solve for the initial fiber stretches using $\hat{\lambda}_p^{-1} \mathbf{a}_0 = \hat{\mathbf{F}}_p^{-1} \cdot \hat{\mathbf{a}}_p$.

At iteration *k* the effective fiber stretch λ_{eff} is determined by,

$$\lambda_{eff}^k \mathbf{a}^k = \mathbf{F}^k \mathbf{F}_p^k \cdot \mathbf{a}_0 . \tag{6.11}$$

Again, keeping $\mathbf{F}^k$ fixed during the update phase, the initial fiber stretch can be enforced by setting $\lambda_e^{k+1} = \hat{\lambda}_p$. Using this as well as $\mathbf{F}_p^{k+1} = \Delta \mathbf{F}_p^{k+1} \hat{\mathbf{F}}_p$, and eliminating the unknown quantity $\mathbf{a}_0$ using Eq. (6.10), Eq. (6.11) can be rewritten as follows.

$$\mathbf{a}^k = \mathbf{F}^k \Delta \mathbf{F}_p^{k+1} \cdot \hat{\mathbf{a}}_p \tag{6.12}$$

Note that this also eliminates $\hat{\lambda}_p$. Although we believe that Eq. (6.12) is the most general way to write an update rule that enforces a given fiber stretch, it is not sufficient to define the update rule uniquely and further assumptions must be made. In the remainder of this section it is assumed that the update rule for $\Delta\mathbf{F}_p^{k+1}$ is such that the orientation of the fiber vectors in the initial reference configuration are unaltered. In other words,

$$\lambda_p^{k+1}\hat{\mathbf{a}}_p = \Delta\mathbf{F}_p^{k+1} \cdot \hat{\mathbf{a}}_p \,. \tag{6.13}$$

Secondly, since only materials that are (nearly-) incompressible will be considered in this manuscript it is required that $\det\left(\Delta\mathbf{F}_p^{k+1}\right) = 1$. A form of $\Delta\mathbf{F}_p^{k+1}$ that accomplishes all of this is given by

$$\Delta\mathbf{F}_p^{k+1} = \lambda_p^{k+1}\hat{\mathbf{a}}_p \otimes \hat{\mathbf{a}}_p + \lambda_2^{k+1}\hat{\mathbf{a}}_2 \otimes \hat{\mathbf{a}}_2 + \lambda_3^{k+1}\hat{\mathbf{a}}_3 \otimes \hat{\mathbf{a}}_3 \,. \tag{6.14}$$

Here, $\hat{\mathbf{a}}_p, \hat{\mathbf{a}}_2, \hat{\mathbf{a}}_3$ form a right-handed coordinate system and $\lambda_2^{k+1}, \lambda_3^{k+1}$ are such that $\lambda_p^{k+1}\lambda_2^{k+1}\lambda_3^{k+1} = 1$. This does not determine $\lambda_2^{k+1}, \lambda_3^{k+1}$uniquely, and a final assumption is necessary, namely that $\lambda_2^{k+1} = \lambda_3^{k+1} = \left(\lambda_p^{k+1}\right)^{-1/2}$. Next, by choosing $\lambda_p^{k+1} = \left(\lambda^k\right)^{-1}$ where λ^k is the fiber stretch induced by $\mathbf{F}^k$ (i.e. $\lambda^k\mathbf{a}^k = \mathbf{F}^k \cdot \hat{\mathbf{a}}_p$). The resulting update rule (using $\mathbf{F}_p^{k+1} = \Delta\mathbf{F}_p^{k+1}\hat{\mathbf{F}}_p$) becomes

$$\mathbf{F}_p^{k+1} = \left[\left(\lambda^k\right)^{-1}\hat{\mathbf{a}}_p \otimes \hat{\mathbf{a}}_p + \left(\lambda^k\right)^{1/2}\left(\mathbf{I} - \hat{\mathbf{a}}_p \otimes \hat{\mathbf{a}}_p\right)\right]\hat{\mathbf{F}}_p \,. \tag{6.15}$$

This can be written more concisely by constructing an orthogonal tensor $\mathbf{Q}$ of which the columns are formed by the vectors $\hat{\mathbf{a}}_p, \hat{\mathbf{a}}_2, \hat{\mathbf{a}}_3$. In that case, Eq. (6.15) can be rewritten as

$$\mathbf{F}_p^{k+1} = \mathbf{Q}\begin{bmatrix} \lambda^{-1} & & \\ & \lambda^{1/2} & \\ & & \lambda^{1/2} \end{bmatrix}\mathbf{Q}^T\hat{\mathbf{F}}_p, \tag{6.16}$$

where the superscript k on λ^k was dropped for clarity. It can easily be verified that this particular update rule satisfies Eq. (6.12) and thus effectively enforces the given fiber stretches. Note that in this case however, the distortion of the geometry is not eliminated, so the reference geometry is altered. This is necessary since the initial reference geometry was not compatible with the given fiber stretches and a new geometry was sought that is compatible.

6.3.2.3. Defining prestrain in terms of fiber stretches

Aside from providing an update rule for $\mathbf{F}_p$, the user must also provide an initial estimate of the prestrain gradient. Although a full 3x3 tensor can be provided, the framework allows the user to express $\hat{\mathbf{F}}_p$ in terms of other quantities. This is useful in the case $\hat{\mathbf{F}}_p$ is constructed from experimental data when knowledge of the full tensor may not be available. For example, consider the case where the fiber stretches $\hat{\lambda}_p$ along the material fibers were determined from experiment. The selection of the form for $\hat{\mathbf{F}}_p\left(\hat{\lambda}_p\right)$ follows the approach in Weiss et al.. It is assumed that the material is transversely isotropic and that the fibers carry most of the load, which implies that the stress in the material is determined predominantly from the fiber stretches $\hat{\lambda}_p$. Furthermore, it is assumed that the material is incompressible and thus the deformation induced by $\hat{\mathbf{F}}_p$ must be isochoric (i.e.

$\det\left(\hat{\mathbf{F}}_p\right)=\hat{J}_p=1$). These assumptions are not sufficient to determine $\hat{\mathbf{F}}_p$ uniquely (in fact, in general the prestrain gradient cannot be determined uniquely), but we argue that, if the stress of the material is determined mostly by $\hat{\lambda}_p$ and $\hat{J}_p$, then *any* form of $\hat{\mathbf{F}}_p$ that results in the given fiber stretches $\hat{\lambda}_p$ and whose determinant satisfies $\hat{J}_p=1$, results in the *same* prestress and is thus a valid prestrain gradient. Thus, given the fiber stretches $\hat{\lambda}_p$ and the fiber vectors in the prestressed reference configuration $\hat{\mathbf{a}}_p$, a form of $\hat{\mathbf{F}}_p$ that satisfies these requirements is given by,

$$\hat{\mathbf{F}}_p=\hat{\lambda}_p\hat{\mathbf{a}}_p\otimes\hat{\mathbf{a}}_p+\hat{\lambda}_p^{-1/2}\left(\mathbf{I}-\hat{\mathbf{a}}_p\otimes\hat{\mathbf{a}}_p\right). \tag{6.17}$$

Using similar arguments as presented in Section 2.2.2, this can be rewritten more concisely using

$$\hat{\mathbf{F}}_p=\mathbf{Q}\begin{bmatrix}\hat{\lambda}_p & & \\ & \hat{\lambda}_p^{-1/2} & \\ & & \hat{\lambda}_p^{-1/2}\end{bmatrix}\mathbf{Q}^T, \tag{6.18}$$

where $\mathbf{Q}$ is an orthogonal tensor whose columns are formed from the vectors $\hat{\mathbf{a}}_p,\hat{\mathbf{a}}_2,\hat{\mathbf{a}}_3$ that define a right-handed coordinate system.

Combining this particular form of $\hat{\mathbf{F}}_p$ (Eq. (6.18)) and the update rule described in Section 2.2.1. (Eq. (6.16)) essentially recovers the method by Weiss et al. which has been used successfully used to enforce the experimental fiber stretches. This application of the framework is illustrated in the example of Section 6.3.4.

6.3.3 Compatibility and recovery of the displacement field

As noted above, the prestrain gradient approach does not rely on the existence of a stress-free configuration. However, given a prestrain gradient field, we can test whether a corresponding stress-free configuration exists, and if it does, it can be recovered.

If $\mathbf{F}_p$ is in fact the gradient of a deformation map, then it must satisfy the following conditions:

$$\det \mathbf{F}_p > 0 \ , \tag{6.19}$$

$$\nabla \times \mathbf{F}_p = \mathbf{0} \ . \tag{6.20}$$

The first condition ensures that $\mathbf{F}_p$ corresponds to a physically possible state, and it must be true even if $\mathbf{F}_p$ is not the gradient of a deformation map. The second condition will only be true if $\mathbf{F}_p$ is the gradient of a deformation map. In other words, if $\mathbf{X} = \eta(\mathbf{y})$, then $\mathbf{F}_p = \nabla_{\mathbf{y}} \eta$. Thus, Eq. (6.20) can be used to determine if the prestrain gradient field is in fact the derivative of a deformation map.

The following algorithm recovers the displacement field from $\mathbf{F}_p$ for the case when it is the derivative of a deformation map. Assume the displacement field $\mathbf{u}(\mathbf{y})$ from the (unknown) stress-free configuration is written in terms of the shape functions in a finite element formulation:

$$\mathbf{u}(\mathbf{y}) = \sum_a N_a(\mathbf{y}) \mathbf{u}_a \ . \tag{6.21}$$

Here, N_a are the shape functions of node *a* and $\mathbf{u}_a$ are the values of the nodal displacements. Inside an element *e*, the gradient of this displacement field at node *b* of element *e* is given by

$$\nabla_{\mathbf{y}}^{e}\mathbf{u}\left(\mathbf{y}_b\right)=\sum_{a}\mathbf{u}_a\otimes\nabla_{\mathbf{y}}^{e}N_a\left(\mathbf{x}_b\right). \tag{6.22}$$

Since the shape function derivatives are in general not continuous across element boundaries, Eq. (6.22) evaluated at the same node *b*, but from a different element will yield a different result. Thus, in the following the value of all elements that contain node *b* must be considered. The set of all elements that connect to node *b* is referred to as the *star of b* and denoted by $S_b=\left\{e\,|\,b\in e\right\}$. Then, the least-squares error at node *b* is defined as follows:

$$E_b=\frac{1}{2}\sum_{e\in S_b}\left\|\nabla^{e}\mathbf{u}\left(\mathbf{y}_b\right)-\mathbf{G}\left(\mathbf{y}_b\right)\right\|_F^2 . \tag{6.23}$$

Here, the sum is taken over the star of node *b* and, $\left\|\bullet\right\|_F$ is the Fröbenius norm and $\mathbf{G}=\mathbf{F}_p-\mathbf{I}$. The *total* error is then defined by summing over all the nodes. Using Eq.(6.22), this can be written as follows:

$$E=\frac{1}{2}\sum_{b}\sum_{e\in S_b}\left\|\sum_{a\in e}\mathbf{u}_a\otimes\nabla^{e}N_a\left(\mathbf{y}_b\right)-\mathbf{G}\left(\mathbf{y}_b\right)\right\|_F^2 . \tag{6.24}$$

By solving this least squares problem, a set of linear equations is recovered that can readily be solved for the nodal displacements $\mathbf{u}_a$. See the Appendix for details.

6.3.4 Implementation aspects

The framework described above was implemented in the FEBio finite element software (Maas et al.) as a plugin, which is an extension to FEBio that is developed and maintained independently from the main FEBio source code. The prestrain plugin defines a new material (called *prestrain elastic*) that is a wrapper around the FEBio elastic materials. The purpose of this material is twofold: to allow the user to define the initial guess for the prestrain gradient $\hat{\mathbf{F}}_p$, and to pass the total elastic deformation gradient $\mathbf{F}_e$ to the functions that calculate the stress and elasticity tensors. FEBio evaluates these tensors in the spatial frame directly using Eq. (6.2). Unlike other approaches that are based on an iterative method for obtaining the prestrain gradient, this approach does not require any change to the existing FEBio source code. In fact, the code that evaluates the constitutive response does not know if the model is being analyzed for a standard forward analysis or a prestrain analysis. Any of the available constitutive (hyperelastic) models in FEBio can be used without modification. This is in part made possible by the ability to define material data hierarchically in FEBio. New element data (in this case the prestrain gradients $\hat{\mathbf{F}}_p$ and $\mathbf{F}_p$) can be defined without changing any existing data structures.

The initial prestrain gradient $\hat{\mathbf{F}}_p$ can be defined as constant over the entire domain, it can be defined for each element, or it can be defined via the fiber stretch. The prestrain plugin can also be expanded and users can create new ways to define the initial prestrain gradient (for instance, a mathematical expression representing the solution of an idealized geometry). The iterative update algorithm described in Section 2.2 is implemented as a nonlinear constraint and accomplishes its goal by using the augmentation mechanism in

FEBio. This mechanism allows constraints to evaluate and modify the state of the model at the end of each converged time step. If a constraint modifies the state of the model, then the time step is re-evaluated. This mechanism is often used in FEBio to determine Lagrange multipliers using the augmented Lagrangian method; in this case it is used to update the prestrain gradient. During the augmentation phase, the prestrain gradient is updated and the convergence is checked. If convergence is met, the augmentation loop finishes and the time step ends. The default implementation does not provide a specific update rule (the base class is an abstract class) and specific update rules are implemented by subclassing the base class. The two update rules discussed in Sections 6.2.2.1 and 6.2.2.2 are available as part of the plugin distribution and can be used as examples that show how to define new update rules.

In general, applying prestrain using this plugin requires a two-step analysis. In the first step, the initial prestrain gradient is given and, if necessary, the update algorithm is applied to find a compatible prestrain gradient. Then, subsequent loading of the model is applied in the second step during which the prestrain gradient is held fixed. The prestrain plugin, associated documentation and example problems are available for download at the FEBio web page (www.febio.org/plugins).

6.4 Results

6.4.1 Verification analysis

This example demonstrates that in the case of a compatible prestrain gradient, Eq. (6.2) indeed results in the correct Cauchy stress. This example uses a hexahedral mesh of a tensile test specimen with dogbone geometry. A transversely isotropic hyperelastic constitutive model was used (Weiss, Maker and Govindjee) - material “coupled trans-iso

Mooney-Rivlin" in FEBio (Maas et al.). The strain energy is defined as:

$$W = c_1\left(I_1 - 3\right) + c_2\left(I_2 - 3\right) + W_f\left(\lambda\right) + U\left(J\right) \tag{6.25}$$

Here, I_1 and I_2 are the first and second invariants of the right Cauchy-Green tensor $\mathtt{C}$ and J is the determinant of the deformation gradient. The fiber stretch λ is given by $\lambda^2 = \mathbf{a}_0 \cdot \mathbf{C} \cdot \mathbf{a}_0$. The strain energy function for the fibers, W_f, is more easily expressed in terms of its derivative with respect to the fiber stretch, but grows exponentially as a function of the fiber stretch. See Weiss, Maker and Govindjee for more details. The term $U\left(J\right) = \frac{1}{2} k \ln\left(J\right)^2$ is the volumetric strain energy function and *k* is the bulk modulus. The material parameters were c_1=1.44 MPa, c_2=0 MPa, c_3=0.57 MPa, c_4=48, c_5=467.1 MPa, *k*=1000 MPa and are representative for MCL ligament (Gardiner, Weiss and Rosenber). The fiber direction was aligned along the long axis of the model.

First, a forward FE analysis was performed, which applied a 10% uniaxial stretch to the sample along the long axis. This was followed by a shear of 10% in the short axis direction (Fig. 6.3, panel A). Then, the nodal coordinates and the deformation gradients were extracted from the forward analysis at the end of the uniaxial stretch and taken as the starting geometry for the prestrain analysis. The deformation gradient of the forward model was used as the initial prestrain gradient of the prestrain analysis. The prestrain model then underwent the same 10% shear. Eq. (6.2) was used for evaluating the stress and spatial tangents in the prestrain analysis. The predicted stresses for the prestrain analysis were nearly identical to those from the forward analysis (Fig. 6.3, panel B). The maximum error in the effective stress was 4x10^{-5} MPa. As a point of reference, the maximum effective

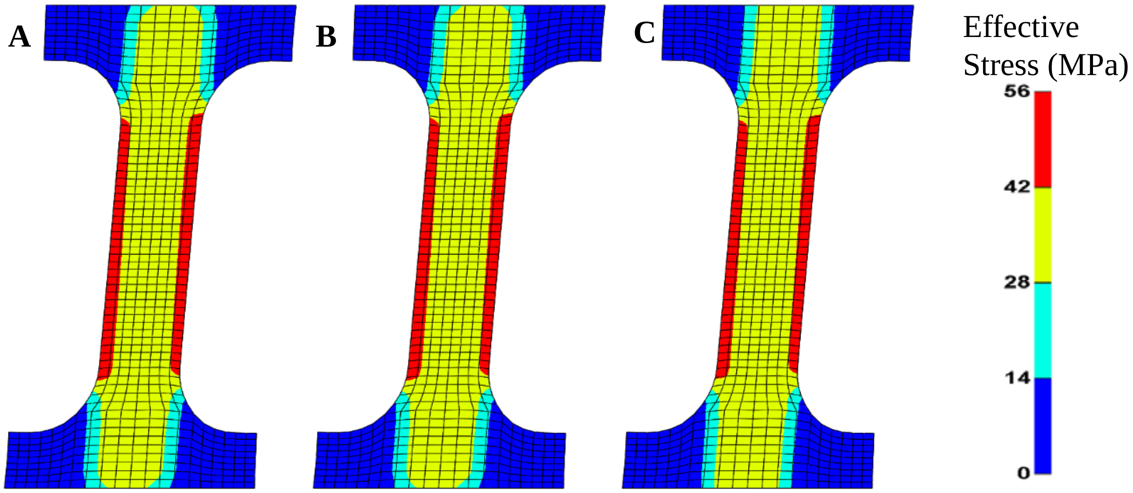


Fig. 6.3: Verification example of the prestrain gradient approach for the case where the prestrain gradient is kinematically compatible. (A) In the forward model, a uniaxial stretch is applied, followed by a shear. The plot shows the effective (von Mises) stress at the end of the analysis. (B) Result for the prestrain analysis where the deformation gradient and nodal coordinates at the end of the uniaxial stretch were taken as the inputs. The prestrain analysis then underwent the same shear step. (C) A similar analysis was performed but using only the fiber stretches instead of the entire deformation gradient from the forward analysis. Again the shear is applied and the effective stress is shown at the end of the shear step.

stress for both models was 54 MPa.

Similarly, the nodal coordinates and the fiber stretches were taken from the forward analysis at the end of the uniaxial stretch and used as the input of an *in situ* stretch model. The *in situ* stretch model then underwent the same 10% shear, using Eq. (6.18) to evaluate the initial prestrain gradient. Since only the fiber stretches were given, the iterative procedure described in Section 6.2.2.2 was used at the start of the analysis to find a compatible prestrain gradient. Again the effective stresses were compared to the results of the forward analysis (Fig. 6.3, panel C). Without the iterative update procedure, the *in situ* strain model resulted in a maximum error of 8 MPa and a maximum effective stress of 53 MPa. The largest errors occurred near the top and bottom of the model geometry. With the iterative procedure, the results improved slightly: the error in the effective stress reduced to 7.4 MPa and the maximum effective stress was 53.2MPa.

6.4.2 Incompatible strain field

This example demonstrates the outcome of applying a prestrain field that is incompatible with the reference configuration. The geometry, representing a medial collateral ligament (MCL) model, and the material properties were taken from published experimental data (Gardiner, Weiss and Rosenber). The MCL was discretized using hexahedral elements and the material was represented using the same transversely-isotropic Mooney-Rivlin constitutive model in Section 3.1 (Eq. (6.25)). The material properties were c_1=4.6 MPa, c_2=0 MPa, c_3=2.4 MPa, c_4=30.6, c_5=323.7 MPa, and k=1,000 MPa.

A heterogeneous *in situ* fiber stretch field was obtained by first applying a prescribed displacement of approximately 5% of total length to the undeformed MCL geometry. In order to simulate experimental uncertainty, the fiber stretch field was smoothed using six iterations of a Laplacian smoothing algorithm (Fig. 6.4, panel A).

The deformed geometry and smoothed fiber stretches from this forward analysis were then used as the starting point of the prestrain analysis. In the prestrain analysis, the ends of the MCL were held fixed and no external loads were applied. Since the applied prestrain was incompatible with the prestressed reference geometry, the mesh distorts and the effective fiber stretches were altered. Visual inspection of the effective fiber stretch (Fig. 6.4, panel B) shows the qualitative difference compared to the applied fiber stretches. The error in the *in situ* stretch was measured by taking the absolute value of the difference:

$$\text{in situ error} = \left| \hat{\lambda}_p - \lambda_{eff} \right| . \tag{6.26}$$

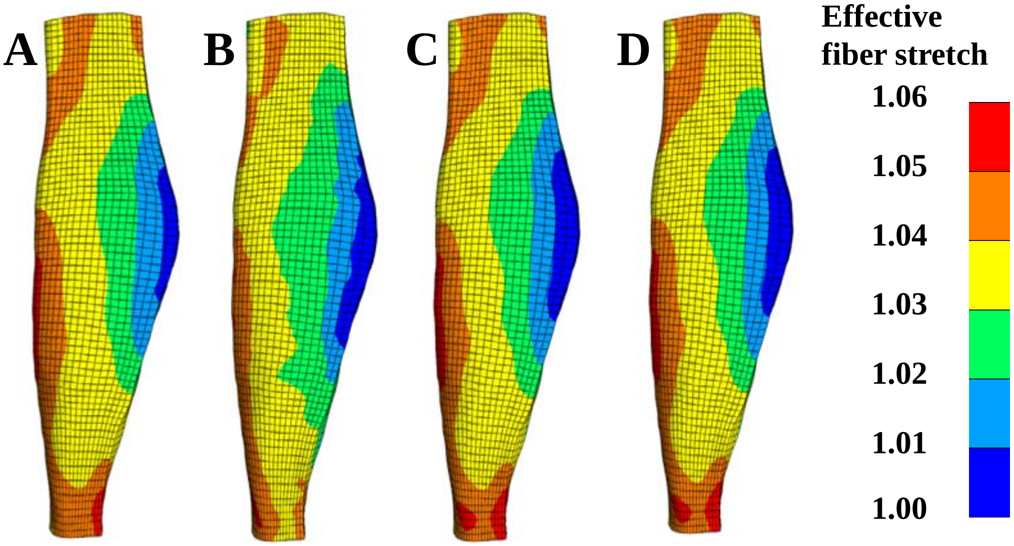


Fig. 6.4: Effects of applying *in situ* stretch to a reference geometry that is kinematically incompatible. (A) Reference geometry with target *in situ* fiber stretch. (B) Deformed state and effective fiber stretch without updates. (C) Effective fiber stretch after eliminating distortion. (D) Effective fiber stretch after enforcing the target *in situ* stretch.

Here, the effective fiber stretch is defined via $\lambda_{eff}\mathbf{a} = \mathbf{F}\hat{\mathbf{F}}_p \cdot \hat{\mathbf{a}}_p$, with $\mathbf{F}$the deformation gradient of the distortion and $\hat{\mathbf{a}}_p$ the fiber vectors in the initial reference configuration. The *in situ* error ranged over the entire mesh between 0 and 0.014.

The distortion of the mesh was quantified by inspecting the effective Lagrange strain E_{eff} ,

$$E_{eff} = \sqrt{\frac{3}{2}\tilde{\mathbf{E}}:\tilde{\mathbf{E}}} \tag{6.27}$$

where $\tilde{\mathbf{E}} = \mathrm{DEV}(\mathbf{E}) := \mathbf{E} - \frac{1}{3}\left(\mathrm{tr}(\mathbf{E})\right)\mathbf{I}$ and calculated from the deformation gradient of the mesh distortion. The observed distortion ranged between 0 and about 0.136. The largest errors occurred near the insertion sites.

6.4.3 Elimination of distortion

For this example the same geometry and *in situ* fiber stretch data were taken from Section 6.3.2. The application described in Section 6.2.2.1 was then applied in order to eliminate the distortion induced by the incompatibility of the prestrain field. As a consequence, the effectively applied prestrain will be different than the initial prestrain field. Qualitatively the effective fiber stretch is noticeably different from the applied fiber stretches (Fig. 6.4, panel C). The maximum *in situ* error, evaluated using Eq. (6.26), grew slightly to 0.016 compared to the example of Section 6.3.2. The maximum distortion error measured over the entire mesh was reduced to 1.24E-3 and was effectively eliminated.

6.4.4 Enforcement of a prestrain deformation gradient field

This example used the same MCL model as Section 6.3.2., but this time the initial heterogeneous *in situ* fiber stretches were enforced using the algorithm in Section 2.2.2. Visual comparison of the resulting effective fiber stretches shows qualitatively the excellent agreement with the applied fiber stretches (Fig. 6.4, panel D). The maximum error in the *in situ*, evaluated by Eq. (6.26), and measured over the entire mesh was reduced to 0.0039. However, the mesh distortion increased to 0.193.

6.4.5 Recovering displacement map from deformation gradient

Using the dogbone mesh from the verification example in Section 6.3.1., the final, deformed geometry and the final deformation gradient were used as the starting point for recovering the original, reference geometry. The deformation gradient from the forward analysis was inverted since the goal was to find the displacement map from the deformed to the reference geometry. Then, the least-squares method described in Section 2.3 was applied to the deformed geometry and the displacement field was recovered. The original reference geometry from the dogbone model was effectively recovered. The maximum displacement of the forward analysis was 1.415 mm and the maximum displacement of the inverse method was 1.405 mm with a maximum relative error of 0.007 (Fig. 6.5). The error is most likely caused by the post-processing step that mapped the deformation gradients from the integration points of the forward analysis to the nodal quantities needed by the inverse method.

6.4.6 Open Knee Example

In the final example, a given *in situ* fiber strain was applied on the ligaments of the Open Knee computational model of the knee joint (Erdemir and Sibole; Sibole et al.;

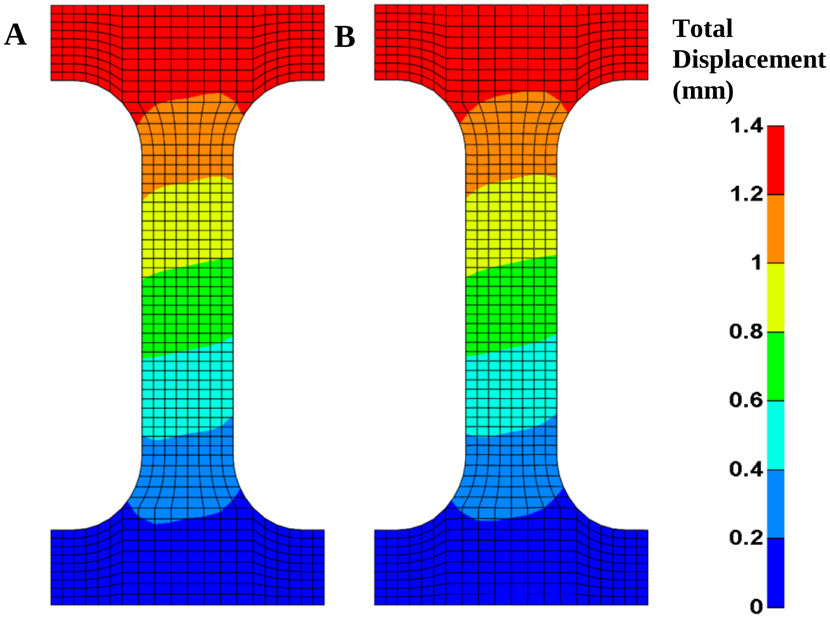


Fig. 6.5: Example illustrating the method to recover the displacement map from the deformation gradient. (A) The reference geometry of the model used in the verification example. (B) The geometry recovered by taking the final, deformed geometry of the verification example and the deformation gradient field and then applying the inverse least-squares method described in the methods section. The fringe plot corresponds to the total displacement (in mm).

Erdemir). The Open Knee model contains the femur and tibia, both assumed rigid, femoral and tibial cartilage, the menisci, and the four major knee ligaments (MCL, ACL, PCL, and LCL). The geometry was meshed with hexahedral elements. Springs are used to connect the meniscal horns to the tibial plateau. The ligaments were modeled using a transversely isotropic material with an isotropic Mooney-Rivlin matrix. The fiber orientations were defined along the long-axis mesh lines. The menisci were modeled with a Fung orthotropic material and the cartilage layers were represented with a Mooney-Rivlin material. Detailed information regarding material coefficients and constitutive models can be found in the Open Knee User's Guide (https://simtk.org/home/openknee). All four cruciate ligaments were prestrained by applying *in situ* strain data taken from published experimental data (Dhaher, Kwon and Barry) and mapped onto the four ligaments. The loading scenario considered was valgus loading to a torque of 10 Nm. The resulting torque on the femur as a function of valgus rotation was compared between the prestrained model and the model without prestrain applied. Similarly, the averaged contact force on the lateral condyle and the effective fiber strains on the MCL were also compared.

Application of prestrain to the knee model (Fig. 6.6), resulted in a much stiffer joint torque-rotation response under valgus loading than the model without prestrain. This can be seen by comparing the valgus torque as a function of valgus rotation (Fig. 6.6, panel B). Similarly, the average contact force over the lateral condyle grew more quickly in the prestrained model (Fig. 6.6, panel D). When compared at a specific angle of valgus rotation (red box in Fig. 6.6, panel B), the predicted fiber strains on the MCL (the primary restraining ligament to valgus rotation in the knee) much higher in the prestrained model (Fig. 6.6, panel C).

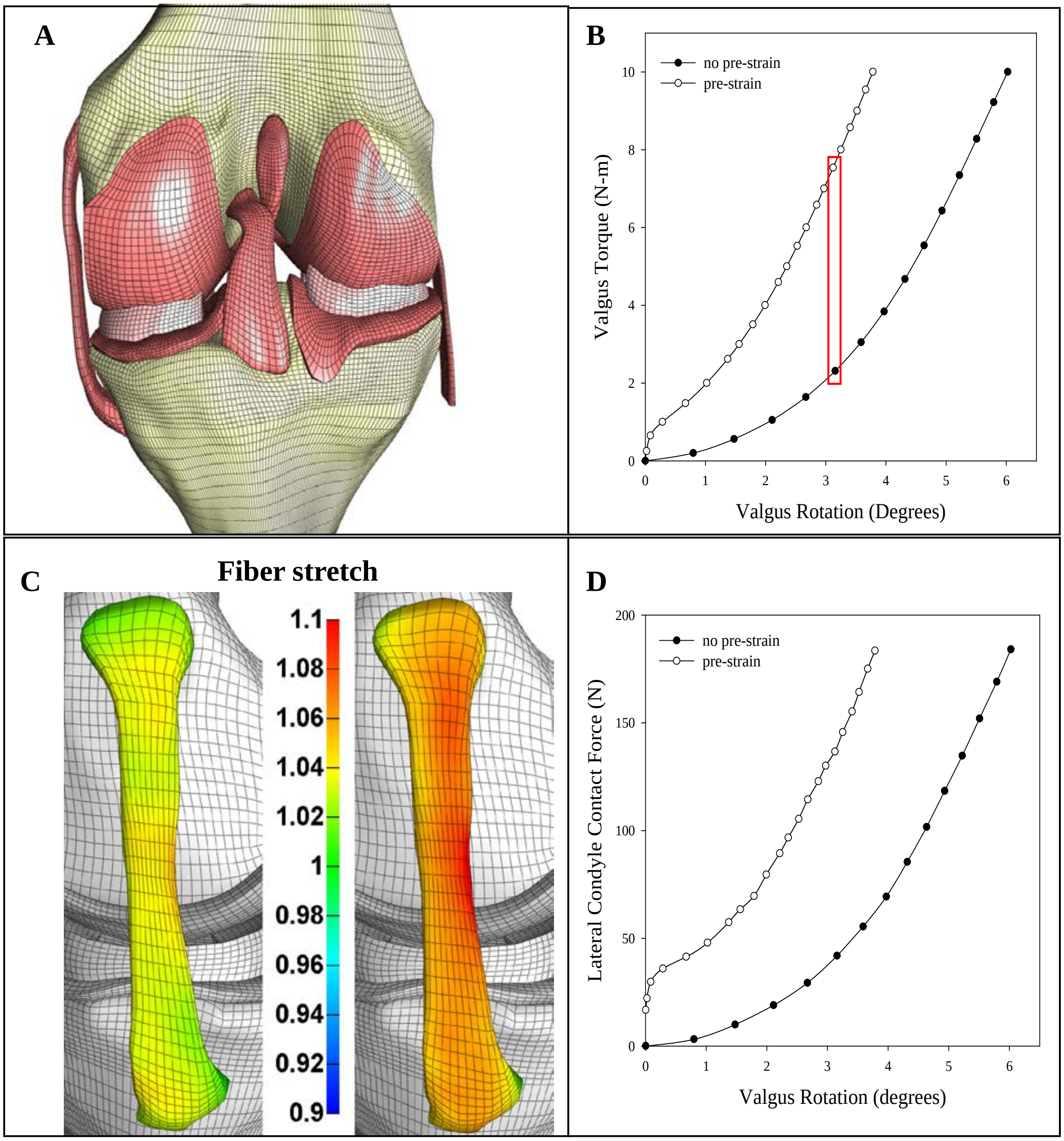


Fig. 6.6: Finite element analysis of the Open Knee model under 10 Nm of valgus torque. A) The Open Knee finite element model, including articular cartilage, menisci, and the four major knee ligaments. B) Valgus torque as a function of valgus rotation for both the prestrained model and the model without prestrain. The red box shows the two points where the fiber strains are compared in panel C. C) Effective fiber stretch at ~3.1 degrees valgus rotation. Left, without prestrain. Right, with prestrain. D) The contact reaction force across the lateral condyle. Results demonstrate that the structural behavior of the joint and predicted ligament strains are dramatically different without prestrain field to the ligaments, and in particular the response is too soft under valgus torque without prestrain.

6.5. Discussion

A general framework was developed to apply prestrain (and therefore prestress) to finite element models in the context of simulating the mechanics of biological structures. The method can solve prestrain problems in a “traditional” manner, where the prestrain deformation gradient field is known (e.g. from analytical solution or recovered from a forward analysis of a global stress-free configuration). In addition, in the case where the applied prestrain field is unknown or just an initial guess, the framework can be used to recover a prestrain gradient field that is compatible with the (possibly altered) reference geometry. Definition of the initial guess and the update rule for the prestrain field are controlled by the user and result in different applications of the framework. Two possible applications were illustrated for the case of applying *in situ* fiber stretches to a model of a MCL. In the first application, the distortion of the mesh was eliminated. In the second application, the distortion was retained and used to find an alternative reference configuration in which the given fiber stretches could be enforced exactly. These two applications demonstrate the great practical use of the proposed framework in applying prestrain.

Although the framework was illustrated in the context of applying fiber stretches by choosing a particular functional form for the prestrain gradient, other forms (e.g. a prestrain field that is the analytical solution of a simplified geometry) can easily be used with the framework. Similarly, other update rules for the prestrain gradient can be implemented easily. However, the two particular update rules considered in Section 6.3.2 reflect the two ends of the spectrum in terms of applying an incompatible prestrain field: either find an alternative prestrain field that eliminates the distortion or enforce the desired prestrain field in an altered reference geometry. The choice of which path to take depends on the goals of the analyst. It can be influenced by the estimated error in the reference geometry compared to the estimated error in the prestrain field. For instance, if the geometry is carefully segmented from high-resolution image data, and thus represents a high-fidelity model of the geometry, it might be important to reduce the induced distortion and to retain the unaltered reference geometry. Alternatively, if the distortion of the reference geometry is small, it may be desirable to enforce the prestrain field without alteration. It might be tempting to ask for both, i.e. can a given prestrain field be enforced on the reference geometry? The answer is “no” in general, unless the applied prestrain field is compatible with the reference geometry in the first place. If the prestrain field is not compatible with the reference geometry, the discrepancies need to be compensated by adjusting either the prestrain field, or the reference geometry, or both as in the case where the prestrain gradient is not updated. The latter case could make interpretation of the results difficult.

It is important to point out that, in general, enforcing a given prestrain field by finding an altered reference geometry (as discussed in Section 6.3.2.2) is an ill-posed inverse problem. The solution may not be unique or worse, may not exist. This poses challenges to applying experimentally obtained prestrain data to a finite element model. If the experimentally obtained prestrain data is too sparse, or the variance in the experimental data is too large, the enforcement of this prestrain may not be possible. On the other hand, the failure of the algorithm in Section 6.3.2.2 to converge indicates that the prestrain data is too inaccurate or incomplete and suggests that additional data (e.g. more accurate fiber stretches or measuring the inhomogeneity in the material parameters) is necessary in order to apply the given prestain.

The general framework recovers other methods for applying prestrain that have been proposed in the literature. For example, consider the problem of applying prestress to an *in vivo* model, where the goal is to determine the stresses when only the deformed geometry and the applied loads are known. Methods for solving these type of problems were presented by Gee, Forster and Wall and expanded on by Weisbecker, Pierce and Holzapfel and Grytz and Downs. These methods attempt to find the deformation gradient iteratively, without recovering the stress-free reference configuration. Our framework can similarly be used to solve this type of problem. In fact, the algorithm described in Section 6.3.2.1 effectively recovers such a method if the initial guess for the prestrain field is taken to be the identity tensor. In that case, the recovered prestrain gradient is in fact the effective prestrain gradient that generates the *in vivo* prestress.

A method was also presented for recovering the displacement field from a given deformation gradient, if the deformation gradient satisfies the kinematic compatibility conditions. This can also be seen as a type of inverse finite element method where the deformation gradient is known as well as the deformed configuration and the reference configuration is sought. This can be of interest in studies where a prestrain field is proposed based on an analytical formulation and the stress-free reference configuration is needed for use in a forward FE analysis (e.g. (Rausch and Kuhl)).

To demonstrate the effect of prestrain on the mechanical response of a biological system, the framework was used on an Open Knee model where *in situ* fiber strains were applied to the four knee ligaments. As expected, application of prestrain to the knee ligaments resulted in a stiffer response for the torque-rotation curve, higher strains on the

MCL, and higher contact stresses on the articular cartilage in the lateral compartment under valgus loading. Thus, the *in situ* strains increased the stability of the knee joint. This further demonstrates that the inclusion of prestrain in computational models is often a necessary requirement to obtain reasonable predictions of subsequent loading conditions of interest.

In summary, the framework presented in this manuscript provides a powerful tool to apply prestrain in large-deformation finite element analysis of biological structures. It is especially useful when the reference configuration in which anatomy is acquired may not be a stress-free state and the stress-free state may not be computable. A significant advantage of the framework is the ease with which it can be customized to recover existing methodologies for applying prestrain, while still allowing new approaches for applying prestrain to be explored.

6.6 Acknowledgments

Funding from NIH grants #R01GM083925, R01GM104139 and R01EB015133 is gratefully acknowledged.

6.7 Appendix

This appendix details how to obtain the linear system of equations to solve for the unknown nodal displacement values, given a compatible deformation gradient tensor F and the deformed geometry. The goal is to find the displacement values that minimize the following least squares error *E*:

$$E = \frac{1}{2} \sum_{b} \sum_{e \in S_b} \left\| \sum_{a \in e} \mathbf{u}_a \otimes \nabla^e N_a \left(\mathbf{y}_b \right) - \mathbf{G}\left(\mathbf{y}_b \right) \right\|_F^2 \quad . \tag{1.1}$$

In order to minimize this expression, the derivative with respect to the unknown nodal deformations must be equated to zero:

$$\frac{dE}{d\mathbf{u}_c} = \sum_{b} \sum_{e \supset b,c} \left(\sum_{a \in e} \mathbf{u}_a \otimes \nabla^e N_a \left(\mathbf{y}_b \right) - \mathbf{G}\left(\mathbf{y}_b \right) \right) \nabla^e N_c \left(\mathbf{y}_b \right) = \mathbf{0} \quad . \tag{1.2}$$

This can be rewritten as follows,

$$\sum_{b} \sum_{e \supset b,c} \left(\sum_{a \in e} \mathbf{u}_a \left(\nabla^e N_a \left(\mathbf{y}_b \right) \cdot \nabla^e N_c \left(\mathbf{y}_b \right) \right) - \mathbf{G}\left(y_b \right) \nabla^e N_c \left(\mathbf{y}_b \right) \right) = \mathbf{0} \, . \tag{1.3}$$

Bringing all the known quantities to the right-hand-side results in

$$\sum_{e \supset c} \sum_{a,b \in e} \mathbf{u}_a \left(\nabla^e N_a \left(\mathbf{y}_b \right) \cdot \nabla^e N_c \left(\mathbf{y}_b \right) \right) = \sum_{e \supset c} \sum_{b \in e} \mathbf{G}\left(\mathbf{y}_b \right) \nabla^e N_c \left(\mathbf{y}_b \right) \quad . \tag{1.4}$$

This is a linear system of equations, which can be solved for $\mathbf{u}_a$. In fact, these are three linear systems of equations with the same coefficient matrix:

$$\begin{aligned} \mathbf{A}\mathbf{u}_x &= \mathbf{R}_x, \\ \mathbf{A}\mathbf{u}_y &= \mathbf{R}_y, \\ \mathbf{A}\mathbf{u}_z &= \mathbf{R}_z. \end{aligned} \tag{1.5}$$

Since the coefficient matrix is the same, the LU decomposition of **A** only needs to be computed once to solve for the $\mathbf{u}_a$.

6.8 References

Alastrue, V., et al. "Assessing the Use of the "Opening Angle Method" to Enforce Residual Stresses in Patient-Specific Arteries." *Ann. Biomed. Eng.* 35.10 (2007): 1821-37.

Balzani, D., J. Schroder, and D. Gross. "Numerical Simulation of Residual Stresses in Arterial Walls." *Comp. Mater. Sci.* 39 (2007): 117-23.

---. "Simulation of Discontinuous Damage Incorporating Residual Stresses in Circumferentially Overstretched Atherosclerotic Arteries." *Acta Biomater.* 2.6 (2006): 609-18.

Bols, J., et al. "A Computational Method to Assess the in Vivo Stresses and Unloaded Configuration of Patient-Specific Blood Vessels." *J. Comput. Appl. Math.* 246 (2013): 10-17.

Chuong, C. J., and Y. C. Fung. "On Residual Stresses in Arteries." *J. Biomech. Eng.* 108.2 (1986): 189-92.

Dhaher, Y. Y., T. H. Kwon, and M. Barry. "The Effect of Connective Tissue Material Uncertainties on Knee Joint Mechanics under Isolated Loading Conditions." *J. Biomech.* 43.16 (2010): 3118-25.

Ellis, B. J., et al. "Medial Collateral Ligament Insertion Site and Contact Forces in the Acl-Deficient Knee." *J. Orthop. Res.* 24.4 (2006): 800-10.

Erdemir, A. "Open Knee: A Pathway to Community Driven Modeling and Simulation in Joint Biomechanics." *J. Med. Device* 7.4 (2013): 0409101-409101.

---. "Open Knee: Open Source Modeling and Simulation in Knee Biomechanics." *J. Knee Surg.* (2015).

Erdemir, A., and S. Sibole. "Open Knee: A Three-Dimensional Finite Element Representation of Knee Joint, User's Guide, Version 1.0.0." 2010. Print.

Fung, Y.C., and S.Q. Liu. "Change of Residual Strains in Arteries Due to Hypertrophy Caused by Aortic Constriction." *Circ. Res.* 65 (1989): 1340-49.

Gardiner, J. C., and J. A. Weiss. "Subject-Specific Finite Element Analysis of the Human Medial Collateral Ligament During Valgus Knee Loading." *J. Orthop. Res.* 21.6 (2003): 1098-106.

Gardiner, J.C., J.A. Weiss, and T.D. Rosenber. "Strain in the Human Medial Collateral Ligament During Valgus Loading of the Knee." *Clin. Orthop. Relat. Res.*391 (2001): 266-74.

Gee, M.W., Ch. Forster, and W.A. Wall. "A Computational Strategy for Prestressing Patient-Specific Biomechanical Problems under Finite Deformation." *Int. J. Numer. Method. Biomed. Eng.* 26 (2010): 52-72.

Greenwald, S.E., et al. "Experimental Investigation of the Distribution of Residual Strains in the Artery Wall." *J. Biomech. Eng.* 1994.4 (1997): 438-44.

Grytz, R, and J.C. Downs. "A Forward Incremental Prestressing Method with Application to Inverse Parameter Estimations and Eye-Specific Simulations of Posterior Scleral Shells." *Comput. Methods Biomech. Biomed. Engin.* 16.7 (2013): 768-80.

Guccione, J. M., A. D. McCulloch, and L. K. Waldman. "Passive Material Properties of Intact Ventricular Myocardium Determined from a Cylindrical Model." *J. Biomech. Eng.* 113.1 (1991): 42-55.

Holzapfel, G. A., and R. W. Ogden. "Modelling the Layer-Specific Three-Dimensional Residual Stresses in Arteries, with an Application to the Human Aorta." *J. R. Soc. Interface* 7.46 (2010): 787-99.

Johnson, Byron E., and Anne Hoger. "The Use of a Virtual Configuration in Formulating Constitutive Equations for Residually Stressed Elastic Materials." *J. Elast.* 41.3 (1995): 177-215.

Labrosse, M. R., et al. "Mechanical Behavior of Human Aortas: Experiments, Material Constants and 3-D Finite Element Modeling Including Residual Stress." *J. Biomech.* 42.8 (2009): 996-1004.

Lujan, T. J., et al. "Effect of Acl Deficiency on Mcl Strains and Joint Kinematics." *J. Biomech. Eng.* 129.3 (2007): 386-92.

Lujan, T. J., et al. "Simultaneous Measurement of Three-Dimensional Joint Kinematics and Ligament Strains with Optical Methods." *J. Biomech. Eng.* 127.1 (2005): 193-7.

Maas, S. A., et al. "Febio: Finite Elements for Biomechanics." *Journal of biomechanical* *engineering* 134.1 (2012): 011005.

Omens, J. H., and Y. C. Fung. "Residual Strain in Rat Left Ventricle." *Circ. Res.* 66.1 (1990): 37-45.

Omens, J. H., A. D. McCulloch, and J. C. Criscione. "Complex Distributions of Residual Stress and Strain in the Mouse Left Ventricle: Experimental and Theoretical Models." *Biomech. Model. Mechanobiol.* 1.4 (2003): 267-77.

Pierce, D. M., et al. "A Method for Incorporating Three-Dimensional Residual Stretches/Stresses into Patient-Specific Finite Element Simulations of Arteries." *J. Mech. Behav. Biomed.* 47 (2015): 147-64.

Raghavan, M. L., et al. "Three-Dimensional Finite Element Analysis of Residual Stress in Arteries." *Ann. Biomed. Eng.* 32.2 (2004): 257-63.

Rausch, M. K., and E. Kuhl. "On the Effect of Prestrain and Residual Stress in Thin Biological Membranes." *J. Mech. Phys. Solids* 61.9 (2013): 1955-69.

Sibole, S., et al. "Open Knee: A 3d Finite Element Representation of the Knee Joint." *34th Annual Meeting of the American Society of Biomechanics*. 2010. Print.

Taber, L. A., and S. Chabert. "Theoretical and Experimental Study of Growth and Remodeling in the Developing Heart." *Biomech. Model. Mechanobiol.* 1.1 (2002): 29-43.

Taber, L. A., et al. "Residual Strain in the Ventricle of the Stage 16-24 Chick Embryo." *Circ. Res.* 72.2 (1993): 455-62.

Vossoughi, J., Z. Hedzaji, and F.S. Borris. "Intimal Residual Stress and Strain in Large Arteries." *Bioengineering Conference ASME*. 1993. Print.

Wang, H. M., et al. "A Modified Holzapfel-Ogden Law for a Residually Stressed Finite Strain Model of the Human Left Ventricle in Diastole." *Biomech. Model. Mechanobiol.* 13.1 (2014): 99-113.

Weisbecker, H., D. M. Pierce, and G. A. Holzapfel. "A Generalized Prestressing Algorithm for Finite Element Simulations of Preloaded Geometries with Application to the Aorta." *Int. J. Numer. Methods Biomed.* 30 (2014): 857-72.

Weiss, J. A., et al. "Three-Dimensional Finite Element Modeling of Ligaments: Technical Aspects." *Med. Eng. Phys.* 27.10 (2005): 845-61.

Weiss, J. A., B. N. Maker, and S. Govindjee. "Finite Element Implementation of Incompressible, Transversely Isotropic Hyperelasticity." *Computer methods in applied mechanics and engineering* 135.1-2 (1996): 107-28.

Woo, S. L., et al. "Measurement of Changes in Ligament Tension with Knee Motion and Skeletal Maturation." *J. Biomech. Eng.* 112.1 (1990): 46-51.

CHAPTER 7

SUMMARY AND DISCUSSION

7.1 Summary

This work focused on the development of several modifications and improvements to FEBio, a finite element code that was designed specifically for solving problems in computational biomechanics and biophysics. The main goal of these improvements was to make FEBio a more extendible and versatile finite element software program in general, but more specifically to facilitate the modeling of problems in computational joint contact mechanics. The ability to accurately model and predict results in this area of research has important applications, ranging from improving our understanding of joint and tissue mechanics, to predicting the outcome of surgical operations.

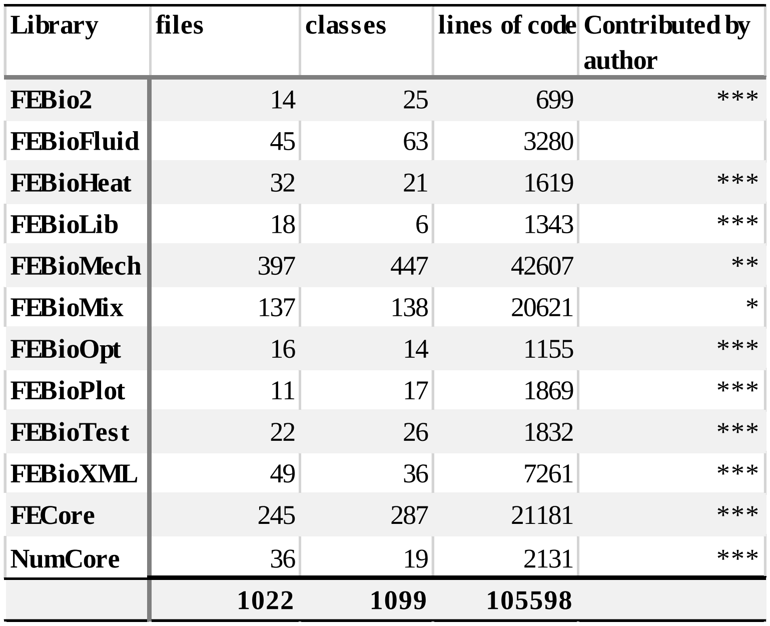
In this final chapter, a brief summary is presented of the three main pillars of this work, namely the development of FEBio's modular structure and its plugin framework, the exploration of quadratic tetrahedral element formulations suitable for large deformation contact analyses, and the ability to include prestrain in models of biological tissues. Some directions for future work are discussed as well.

7.1.1 FEBio development and impact

Since its initial release in 2007, FEBio has grown exponentially, in part due to the work presented here but also due to contributions of others (Ateshian, Albro, et al.; Ateshian, Maas and Weiss; Ateshian, Maas and Weiss; Ateshian, Maas and Weiss;

Ateshian, Nims, et al.; Ateshian, Rajan, et al.; Ateshian and Ricken), and continues to grow to this day. At the time of writing, FEBio is composed of 12 libraries, collecting a total of 1022 files and 1099 classes, and contains roughly 100,000 lines of code (not including comments) (Table 7.1). The FEBio user's community counts at least 5,100 users and the software itself has been downloaded over 110,000 times since its initial release. Many of the FEBio users are students, some new to finite element modeling, but experienced academics and researchers are also part of its user base. FEBio has been used in a significant way in at least 216 peer reviewed journal articles, covering areas of soft tissue mechanics, tissue engineering, joint and muscle mechanics, damage modeling, cardiovascular mechanics, angiogenesis, modeling of vision and hearing, multiscale modeling, and many more (see http://febio.org/publications/journal-articles/ for a complete

Table 7.1. Development effort metrics for the various components of the FEBio software. The last column shows a rough estimate of how much the author contributed to the individual libraries: *** = mostly developed by author, few contributions by others; ** = significant contributions by author, but also by others; * = minor contributions by author; no stars = no contributions by author.



Library	files	classes	lines of code	Contributed by author
FEBio2	14	25	699	***
FEBioFluid	45	63	3280	
FEBioHeat	32	21	1619	***
FEBioLib	18	6	1343	***
FEBioMech	397	447	42607	**
FEBioMix	137	138	20621	*
FEBioOpt	16	14	1155	***
FEBioPlot	11	17	1869	***
FEBioTest	22	26	1832	***
FEBioXML	49	36	7261	***
FECore	245	287	21181	***
NumCore	36	19	2131	***
	1022	**1099**	**105598**	

list). FEBio is also used as part of several courses that are being taught in institutes around the world, including University of Michigan, Boise State University, Technical University of Madrid, National University of Singapore, University of Strathclyde, Brigham Young University, Eindhoven Technical University, Tel Aviv University, McGill University, and more.

These statistics show that FEBio has made a significant and lasting impact in the biomechanics community. Even more, FEBio has grown to be a competitive, and in some areas even superior, alternative to the traditional commercial packages that have been used predominantly in this field. Evidence of this can be found in several recent publications that have compared FEBio to Abaqus, by many considered the de facto standard in computational biomechanics. For instance, two independent studies (Galbusera et al.; Meng et al.) demonstrated that FEBio's biphasic contact modeling compared favorably to Abaqus and both codes were found effective for these types of simulations. Another study (Pierrat et al.) implemented a novel constitutive model for transversely isotropic materials in both Abaqus (as a user subroutine) and FEBio (as a material plugin). It concluded that the results of both codes compared well, although FEBio performed better in terms of run-time, and that the implementation efforts were considerably less in FEBio, mostly thanks to its extensive tensor class and operator library. Another example is the OpenKnee project (Erdemir; Erdemir), a collaborative effort between several institutes that aims, among other things, to make available free, open-access models of the human knee joint. Its decision to move away from providing its models as Abaqus input files and convert them to FEBio input files is further evidence that the community has embraced FEBio as a viable alternative.

7.1.2 Parallelization of FEBio

In most implicit finite element solvers, including FEBio, a significant amount of time is spent in factoring the global stiffness matrix. In a serial implementation, the factorization can take up roughly 75%-95% of the total execution time. Therefore, FEBio has always made use of parallel linear solvers to speed up the factorization process. FEBio implements an abstract interface to the linear solver so that in principle, any linear solver library could be used in FEBio. A specific linear solver can be supported by implementing a proxy class derived from this abstract base class that maps the external library to FEBio's framework. Proxy classes that interface to several existing linear solver packages are provided (e.g. SuperLU, Pardiso, MKL, WSMP) and others can be added easily. The framework supports both direct and iterative linear solvers. With the use of parallel linear solvers, a significant speedup can be obtained on multicore systems. Although the actual speedup depends strongly on the particular model under investigation, speedups up to a factor of 4 are not uncommon on shared memory systems with 8 core, a common architecture on modern desktop computers.

With the use of parallel linear solvers more time is being spent in evaluating the stiffness matrices and right-hand-side vectors. For some complex constitutive models (e.g. EFD materials (Ateshian, Rajan, et al.), computational homogenization (Kouznetsova)), the amount of time evaluating these quantities can be relatively large. As a result, several parts of FEBio itself have been parallelized using OpenMP. This includes the evaluation of the global right-hand-side vector and the global stiffness matrix. These routines typically contain loops over the elements of the mesh, which can be parallelized easily since the corresponding element quantities can be evaluated independently. The only difficulty arises during the assembly of the element quantities in the global structure. To avoid a data collision where several threads attempt to write to the same memory location, a critical section is placed inside the assembly operation where the global structure is accessed. With this simple, yet effective, parallelization strategy, a significant reduction of the runtime was achieved, especially for problems that make use of these complicated constitutive models mentioned earlier.

7.1.3 Quadratic tetrahedral formulations

This work investigated two quadratic tetrahedral element formulations, a 10-node quadratic, and a 15-node "serendipity" quadratic tetrahedron for modeling problems in joint contact mechanics. It explored the effects of several volume and surface integration rules on accuracy and computational cost for various contact problems that were representative of articular contact mechanics. The integration rules that were chosen were such that they integrate polynomials of at least second degree exactly for undistorted tetrahedra. (In Chapter 4, we referred to this property somewhat incorrectly as the "order" of the integration rule. The order of an integration rule usually refers to how fast the error term decreases as the number of sampling points is increased.) In fact, the V4 formulation integrates exactly polynomials of second degree, V8 third degree, V11 fourth degree, and V15 fifth degree (Keast). Similarly, for the surface integration rules, the S3 rule integrates second degree, and S7 integrates fifth degree polynomials exactly (Witherden and Vincent). In addition, these integration rules were all symmetric and, with the exception of the V11 rule, had all positive weights. Although most of the rules considered in this work appeared to be sufficient for obtaining accurate results, exploration of additional, more efficient integral rules may be of interest. Some more efficient rules were recently presented (Witherden and Vincent). For instance, this paper reports a 14 point symmetric integration rule that integrates polynomials of fifth degree exactly, just like the V15 rule used in this work. However, it must be kept in mind that these rules are designed for integrating polynomials exactly. For large deformation contact problems, both the volume integrands and the contact surface integrands are nonlinear functions and therefore these rules may not integrate these functions exactly. This is another reason to continue exploring additional integration rules that add integration points or vary their location to obtain more accurate results.

The super-convergent patch recovery method (Zienkiewicz and Zhu) was used for accurate recovery of the stresses. For the problems considered, the quadratic tetrahedral formulations appeared to be less susceptible to locking than linear tetrahedrons and converged at least as good as hexahedral elements. The results appeared more sensitive to the choice of surface integration rule. Overall, the higher order surface integration rules provided more accurate recovery of the contact stresses. The volume integration rule did not affect the results significantly, however it was found that the combination of the S3 surface rule and the V11 volume rule converged poorly and resulted in inaccurate stresses. In fact, we found that this combination may sometimes lead to instabilities, such as negative Jacobians that prevent the solution from progressing, and may ultimately lead to failure to converge. Consequently, this combination should best be avoided. Despite the insensitivity of the volume integration rule, it might be necessary in some cases to consider the higher-order rules considered in this work to prevent ill conditioning of the least-square approach that is used to recover the stresses. In this work, the runtimes of models were compared based on the number of nodes. Other metrics could have been used (e.g. number of elements), but this metric was chosen as it correlates strongly to the size of the linear system of equations that needs to be solved. Since factoring and solving this linear system of equations is for many problems the most time consuming part of the overall solution process, the number of nodes seemed a fair metric to evaluate overall runtime behavior. Given approximately the same number of nodes (and thus degrees of freedom or equations), it was found that the quadratic tetrahedral formulations appeared to run faster than the other element formulations. These results demonstrated that quadratic tetrahedral formulations are superior to the linear tetrahedral formulations and are viable alternatives to the "gold standard" three-field hexahedral elements in the context of joint contact mechanics.

The advantages offered by quadratic tetrahedral formulations have spurred several new research projects that use FEBio to employ these elements in their models. In Klennert et al., the articular cartilage layers of a human hip joint were modeled with tetrahedral elements, a structure that is usually modeled with hexahedral elements. The purpose of this work is to study the effects of simulated chondral defects and labral delamination on cartilage mechanics. The use of tetrahedral elements greatly facilitated the meshing of the complex geometries, especially the local modifications that were required to simulate the chondral defects. In Phuntsok et al., quadratic tetrahedral elements are used to mesh the cervical spine, a structure that is geometrically very complex due to its many protrusions. As a final example, the Open Lumbar Spine (Finley, Brodke and Ellis) is a project that wishes to make available open-access, validated models of the human lumbar spine. Although initially this project used hexahedral elements for meshing the geometries, it is currently exploring whether quadratic tetrahedral elements offer a good alternative.

7.1.4 Modeling prestrain in biological tissues

Many biological tissues show evidence of prestress and prestrain (Fung and Liu; Gardiner, Weiss and Rosenber; Omens and Fung), stress or strain that exists in the unloaded reference configuration. Although most researchers agree that modeling these prestresses is important for obtaining the accurate mechanical response, including them in finite element models has proven to be challenging. Researchers usually emulated prestrain via a judicious choice of boundary conditions available in closed-source commercial packages (Balzani, Schroder and Gross; Rausch and Kuhl). This approach is severely limited as it requires explicit knowledge of the stress-free reference configuration. Others developed or modified custom finite element codes in order to include prestrain directly (Alastrue et al.; Gee, Forster and Wall; Weisbecker, Pierce and Holzapfel). Although this approach is more versatile, modifications to custom in-house developed codes hamper dissemination and reproducibility of results.

With the prestrain plugin that was presented in Chapter 6, FEBio users are now able to include prestrain in their computational models of biological tissues easily and in a variety of ways. The implemented method is based on a multiplicative decomposition of the total elastic deformation gradient (Gee, Forster and Wall). This tensor, which is evaluated as the product of the deformation gradient and the prestrain gradient tensor, is then used to evaluate the material response. This approach allows prestrain to be applied to any of the hyper-elastic constitutive models available in FEBio. The proposed framework is more general than previously published approaches, and in fact recovers several methods as special cases as was demonstrated in Chapter 6. Two particular applications were demonstrated, namely the enforcement of a user-defined fiber stretch map onto a transversely isotropic model of ligament, and the elimination of mesh distortion of the reference configuration in the case of an incompatible prestrain gradient.

As opposed to some other methods for prestressing the reference geometry (e.g. (Bols et al.)), the proposed method does not recover the stress-free reference configuration. Although in general such a stress-free configuration may not exist or may not be computable if it does exist, there may be situations in which the geometry of the stress-free configuration is desired. For this reason, a novel method was presented that can recover the reference configuration from the given deformed configuration and the deformation map.

7.2 Future Work

7.2.1 FEBio, a multiphysics solver

The specific needs of the biomechanics and biophysics communities continue to be the main driving force behind FEBio's development. Although the initial focus was on solid mechanics, with the implementation of biphasic, multiphasic (Ateshian, Maas and Weiss), electrokinetics, and chemical reactions (Ateshian, Nims, et al.), it has grown into a true multiphysics solver. Due to the importance of solving coupled-physics types of problems in computational biophysics, the further development of FEBio is centered on its multiphysics solution capabilities. In particular, the incorporation of a fluid mechanics solver as well as a fluid-solid-interaction (FSI) solver are two of the main focal points. There are several relevant applications in biomechanics and biophysics that would benefit from such capabilities, such as cardiovascular mechanics, cerebrospinal mechanics, vocal fold and upper airway mechanics, viscous flow over endothelial cells, canalicular and lacunar flow over osteocytes, diarthrodial joint lubrication, etc.

The implementation of these new solution algorithms is greatly facilitated by the improved modular framework developed in this work. However, to support coupled-physics applications optimally, some modifications to FEBio's structure are currently being explored that focus on the implementation of alternative solution strategies, such as staggered solution methods, continuation methods (Leger, Deteix and Fortin), multifield formulations, solution methods based on the Schur-complement, adaptive mesh refinement (Leger and Pepin), etc. In addition, these modifications will allow new solution strategies to be implemented in the plugin framework, which is currently not possible. This would further facilitate the exploration of novel solution strategies for coupled-physics problems with FEBio and greatly expand the applications of the plugin framework. In fact, the ability to implement solution strategies for solving coupled-physics problems as a plugin opens the door to the development of a programmable environment for FEBio. This will allow users to implement new finite element solvers, independently of FEBio, while at the same time taking full advantage of the services offered by FEBio, such as solving linear and non-linear systems of equations, automation of the input file, customization of the output file, and more.

7.2.2 Stress recovery for higher-order elements

In this work, the superconvergent patch recovery (SPR) method (Zienkiewicz and Zhu) was used for the recovery of the stresses. This method was originally proposed as a mechanism for estimating the error in the finite element solution. A posteriori error estimation is an important aspect of finite element analysis. Not only does it provide a way of assessing the accuracy of the solution, but it also is an important ingredient for adaptive mesh refinement strategies.

The SPR method recovers the stresses at the nodes by fitting a polynomial to the sampling points inside the elements where the stresses are evaluated (e.g. integration points). The order of the polynomial that can be used to recover the stresses depends on the number of sampling points. In FEBio, the integration points are used as the sampling points for recovery, which makes the stress recovery sensitive to the integration rule. As noted above, for the TET10 element, we used a linear polynomial to avoid ill-conditioning of the SPR recovery using some of the lower-order integration rules that were implemented for this element. However, an easy way to overcome this limitation would be to enrich the sampling points by adding additional points aside from the integration points. These additional points would then only be used by the stress recovery process. That way, a quadratic polynomial could be used for the TET10 elements, just like was done for the TET15 elements.

Since the inception of the SPR method, several researchers have proposed alternative recovery methods that in some cases are superior to the original Zienkiewicz-Zhu SPR method. (See the review article by Grätsch and Bathe for an overview.) However, many of these methods are focused on recovery of the displacement gradients (e.g.(Zhang and Naga)). It is not straightforward how such methods could be adapted to recover the stresses, since it would require a posteriori evaluation of the constitutive routines, which may not be possible for history-dependent materials.

There are some recently proposed methods that focus on stress recovery and that may offer some advantages over the SPR approach. The main problem with this method is that it relies on the existence of superconvergent points, i.e. points at which the stress is of higher order accuracy than anywhere else in the element domain. However, such points do not always exist, e.g. in curved iso-parametric elements or elements with nonlinear material behavior (Hiller and Bathe). Therefore, methods that do not rely on the existence of superconvergent points may offer a distinct advantage for such elements. In Gu, Zong and Hung, which focuses on nonlinear analyses, the SPR method is modified and it was shown that this modified method obtains good results using the standard Gauss integration points. The method was illustrated on bilinear quadrilateral elements, but could easily be applied to higher-order tetrahedral elements. Another interesting method was proposed by Payen and Bathe and uses the equivalent nodal point forces (NPF) that are evaluated during the assembly of the residual vector. It was observed that these forces are of higher order accuracy than the stresses. The method then tries to find the nodal stresses that in a least-square sense match the nodal point forces. Although the approach has only been applied to linear elements, it could easily be extended to higher-order elements. In Payen and Bathe a general, variational framework was proposed that in special cases recovers several other stress recovery methods, including the NPF method. Using this framework, a simple, yet effective, method is proposed for recovering stresses. Again, the method is only demonstrated on linear elements, but could be extended to higher-order elements. It would be of great interest to investigate whether these new stress recovery methods would be effective for nonlinear, large deformation analyses using higher-order tetrahedral elements.

7.2.3 Higher-order elements in biphasic contact

The quadratic tetrahedral element formulations have proven to be very successful in simulations of joint contact mechanics. The next step would be to investigate whether these element formulations offer distinct advantages in biphasic and multiphasic contact as well. These types of analysis are complicated by the requirement that the effective fluid pressure and effective solute concentrations must be continuous across the contact interface (Ateshian, Maas and Weiss) . Until now, computational models of these types of analyses have predominantly used hexahedral elements. Linear tetrahedral formulations have not been proven very successful, in part, by the fact that biphasic materials act analogously to nearly-incompressible materials, especially during a rapid initial loading phase (Ateshian, Ellis and Weiss). Predictions of linear tetrahedral models would be very inaccurate during such a phase due to their tendency to lock. However, biphasic models often require a fine discretization near surfaces where fluid is allowed to leave in order to capture the rapidly changing pressure gradients. A uniformly refined mesh may slow the computations unnecessarily, so a discretization that is only refined near the surface is preferred. For complex geometries, such locally refined meshes are generally easier to create with tetrahedral meshes. Thus, the use of quadratic tetrahedral elements may offer some advantages over the hexahedral elements in this area of application of joint contact mechanics as well.

Even more, biphasic contact analyses may also benefit from adaptive mesh refinement (AMR) strategies to further automate the mesh generation process. Such techniques, in combination with efficient continuation methods, have proven to be effective for large deformation elasticity problems (Leger and Pepin) and could be extended to biphasic and multiphasic analyses. In this case, the challenge would be to determine reliable heuristics to drive the mesh refinement process. In addition to the displacement gradients, one would need to consider accuracy metrics that include the effective pressure gradients and effective solute concentration gradients as well to drive the refinement phase. Accurate recovery for flux quantities can presumable be obtained with the superconvergent patch recovery method, which was already successfully applied for recovering accurate stresses. Although mesh refinement of hexahedral elements is feasible, in general, mesh refinement strategies assume tetrahedral meshes. This further motivates the exploration of higher-order tetrahedral formulations for biphasic contact analyses.

7.2.4 Inverse FE

One of the applications of the prestrain plugin presented in this work is the solution of certain types of inverse FE problems where the deformed configuration is given, as well as the loads that exist in this configuration. The goal is then to find the stresses that exist in the known deformed configuration. As was shown in Chapter 6, several methods exist for solving these kinds of problem and the prestrain plugin offers a convenient tool for doing so. In this case, the initial prestrain gradient tensor is defined as the identity tensor, and the constraint that eliminates the geometry distortion is applied when the loads are increased to the desired level. The result is a prestrain gradient tensor that represents the deformation gradient from the unknown reference configuration to the known deformed configuration, and that generates the stresses that balance the applied load.

The prestrain approach outlined in Chapter 6 does not recover the original stress-free reference configuration. Although, in general, it cannot be guaranteed that such a stress-free reference configuration exists or is computable, in many applications, such as the inverse problem outlined above, it does. In these cases, it may be useful if the stress-free reference configuration could be recovered as well. A method was presented in Chapter 5 that recovers the stress-free configuration, given the deformed configuration and the deformation gradient. This method does requires the solution of an additional linear system of equations, but this is not seen as a significant computational overhead since the solution of the prestrain problem usually will be computationally more expensive.

However, there are some methods for solving inverse FE methods that do return the reference configuration as part of the solution. One such method of interest is the Backward Displacement Method (BDM) by Bols et al.. This method is appealing since it is relatively simple and treats the forward solver as a black box, so that it could work with any finite element solver. This makes the method an attractive candidate for a FEBio plugin. The method was implemented successfully as a plugin, but the initial results showed poor convergence for problems that contain large rotations. Although Bols provided a way to accelerate convergence, we are currently investigating an alternative way that promises much better results. The method is similar to the BDM, but differs in the step that updates the reference configuration in that it makes explicit use of the approximate deformation gradient. Initial results show that this new method can be applied to problems undergoing large rotations, an area of application where BDM typically fails (Fig. 7.1).

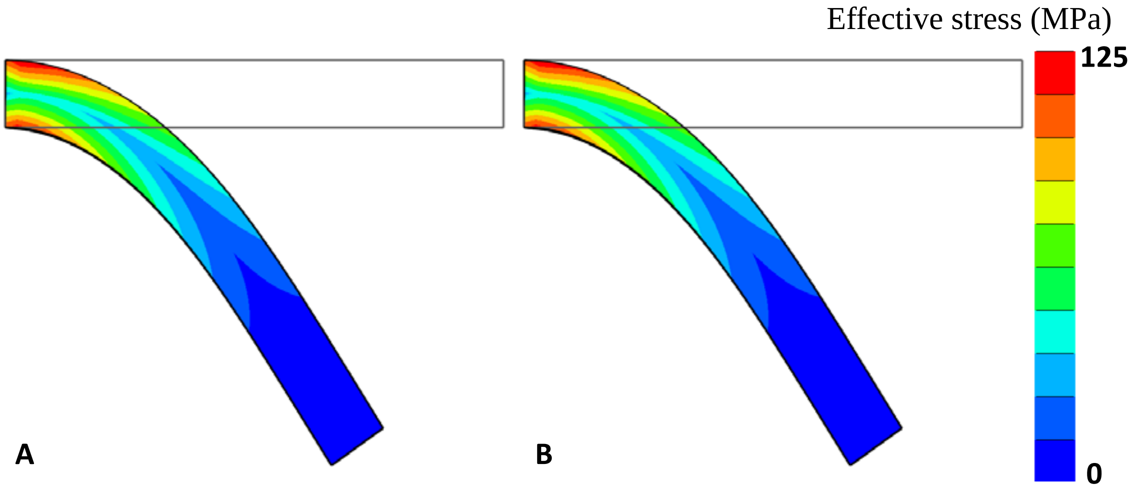


Fig. 7.1. Comparison of forward and inverse FE problem. A) Forward analysis. Pressure applied to top surface deforms a hyper-elastic beam. Color contours indicate effective (von Mises) stress. B) Inverse analysis. Starting from the deformed configuration and known pressure load, the stress as well as the original reference configuration is recovered.

7.3 References

Alastrue, V., et al. "Assessing the Use of the "Opening Angle Method" to Enforce Residual Stresses in Patient-Specific Arteries." *Ann. Biomed. Eng.* 35.10 (2007): 1821-37.

Ateshian, G. A., et al. "Finite Element Implementation of Mechanochemical Phenomena in Neutral Deformable Porous Media under Finite Deformation." *J. Biomech. Eng.* 133.8 (2011): 081005.

Ateshian, G. A., B. J. Ellis, and J. A. Weiss. "Equivalence between Short-Time Biphasic and Incompressible Elastic Material Response." *J. Biomech. Eng.* 129.3 (2006): 405-12.

Ateshian, G. A., Steve A. Maas, and J. A. Weiss. "Finite Element Algorithm for Frictionless Contact of Porous Permeable Media under Finite Deformation and Sliding." *J. Biomech. Eng.* 132.6 (2010).

---. "Multiphasic Finite Element Framework for Modeling Hydrated Mixtures with Multiple Neutral and Charged Solutes." *J. Biomech. Eng.* 135.11 (2013).

---. "Solute Transport across a Contact Interface in Deformable Porous Media." *J. Biomech.* 45.6 (2012): 1023-27.

Ateshian, G. A., et al. "Computational Modeling of Chemical Reactions and Interstitial Growth and Remodeling Involving Charged Solutes and Solid-Bound Molecules." *Biomech. Model. Mechanobiol* 13.5 (2014): 1105-20.

Ateshian, G. A., et al. "Modeling the Matrix of Articular Cartilage Using a Continuous Fiber Angular Distribution Predicts Many Observed Phenomena." *J. Biomech. Eng.* 131.6 (2009): 061003.

Ateshian, G. A., and T. Ricken. "Multigenerational Interstitial Growth of Biological Tissues." *Biomech. Model. Mechanobiol* 9.6 (2010): 689-702.

Balzani, D., J. Schroder, and D. Gross. "Simulation of Discontinuous Damage Incorporating Residual Stresses in Circumferentially Overstretched Atherosclerotic Arteries." *Acta Biomater.* 2.6 (2006): 609-18.

Bols, J., et al. "A Computational Method to Assess the in Vivo Stresses and Unloaded Configuration of Patient-Specific Blood Vessels." *J. Comput. Appl. Math.* 246 (2013): 10-17.

Erdemir, A. "Open Knee: A Pathway to Community Driven Modeling and Simulation in Joint Biomechanics." *J. Med. Device* 7.4 (2013): 0409101-409101.

---. "Open Knee: Open Source Modeling and Simulation in Knee Biomechanics." *J. Knee Surg.* (2015).

Finley, S., D.S. Brodke, and B. J. Ellis. "Open Lumbar Spine: Free, Validated, Nonlinear Finite Element Models of the Human Lumbar Spine." *Comput. Methods Biomech. Biomed. Engin.* In Preparation (2016).

Fung, Y.C., and S.Q. Liu. "Change of Residual Strains in Arteries Due to Hypertrophy Caused by Aortic Constriction." *Circ. Res.* 65 (1989): 1340-49.

Galbusera, F., et al. "Comparison of Various Contact Algorithms for Poroelastic Tissues." *Comput. Methods Biomech. Biomed. Engin.* 17.12 (2014): 1323-34.

Gardiner, J.C., J.A. Weiss, and T.D. Rosenber. "Strain in the Human Medial Collateral Ligament During Valgus Loading of the Knee." *Clin. Orthop. Relat. Res.*391 (2001): 266-74.

Gee, M.W., Ch. Forster, and W.A. Wall. "A Computational Strategy for Prestressing Patient-Specific Biomechanical Problems under Finite Deformation." *Int. J. Numer. Method. Biomed. Eng.* 26 (2010): 52-72.

Grätsch, Thomas, and Klaus-Jürgen Bathe. "A Posteriori Error Estimation Techniques in Practical Finite Element Analysis." *Comput. Struct.* 83.4–5 (2005): 235-65.

Gu, H., Z. Zong, and K. C. Hung. "A Modified Superconvergent Patch Recovery Method and Its Application to Large Deformation Problems." *Finite Elem. Anal. Des.* 40.5–6 (2004): 665-87.

Hiller, Jean-François, and Klaus-Jürgen Bathe. "On Higher-Order-Accuracy Points in Isoparametric Finite Element Analysis and an Application to Error Assessment." *Comput. Struct.* 79.13 (2001): 1275-85.

Keast, P. "Moderate-Degree Tetrahedral Quadrature Formulas." *Comput. Methods Appl. Mech. Eng.* 55.3 (1986): 339-48.

Klennert, B.J., et al. "The Effects of Simulated Focal Chondral Defects and Labrum Delamination on Cartilage Mechanics in a Dysplastic Hip." *J. Biomech.* In 2nd Review (2016).

Kouznetsova, V.G. "Computational Homogenization for the Multi-Scale Analysis of Multi-Phase Materials." Technische Universiteit Eindhoeven, 2002. Print.

Leger, S., J. Deteix, and A. Fortin. "A Moore-Penrose Continuation Method Based on a Schur Complement Appraoch for Nonlinear Finite Element Bifurcation Problems." *Comput. Struct.* 152 (2015): 173-84.

Leger, S., and A. Pepin. "An Updated Lagrangian Method with Error Estimation and Adaptive Remeshing for Very Large Deformation Elasticity Problems: The Three-Dimensional Case." *Comput. Methods Appl. Mech. Eng.* 309 (2016): 1-18.

Meng, Q., et al. "Comparison between Febio and Abaqus for Biphasic Contact Problems." *Proc Inst Mech Eng H* 227.9 (2013): 1009-19.

Omens, J. H., and Y. C. Fung. "Residual Strain in Rat Left Ventricle." *Circ. Res.* 66.1 (1990): 37-45.

Payen, Daniel Jose, and Klaus-Jürgen Bathe. "Improved Stresses for the 4-Node Tetrahedral Element." *Comput. Struct.* 89.13–14 (2011): 1265-73.

---. "A Stress Improvement Procedure." *Comput. Struct.* 112–113 (2012): 311-26.

Phuntsok, Rinchen, et al. "The Occipitoatlantal Capsular Ligaments Are the Primary Stabilizers of the Pediatric Craniocervical Junction." *Spine* In Preparation (2016).

Pierrat, Baptiste, et al. "Finite Element Implementation of a New Model of Slight Compressibility for Transversely Isotropic Materials." *Comput. Method. Biomec.* 19.7 (2015): 1-14.

Rausch, M. K., and E. Kuhl. "On the Effect of Prestrain and Residual Stress in Thin Biological Membranes." *J. Mech. Phys. Solids* 61.9 (2013): 1955-69.

Weisbecker, H., D. M. Pierce, and G. A. Holzapfel. "A Generalized Prestressing Algorithm for Finite Element Simulations of Preloaded Geometries with Application to the Aorta." *Int. J. Numer. Methods Biomed.* 30 (2014): 857-72.

Witherden, F. D., and P. E. Vincent. "On the Identification of Symmetric Quadrature Rules for Finite Element Methods." *Comput. Math. Appl.* 69.10 (2015): 1232-41.

Zhang, Zhimin, and Ahmed Naga. "A New Finite Element Gradient Recovery Method: Superconvergence Property." *SIAM J. Sci. Comput.* 26.4 (2005): 1192-213.

Zienkiewicz, O.C., and J.Z. Zhu. "The Superconvergent Patch Recovery and a Posteriori Error Estimates. Part 1: The Recovery Technique." *Intl. J. Num. Meth. Eng.* 33 (1992): 1331-634.

CHAPTER 8

APPLICATIONS OF FEBIO

8.1 Introduction

The focus of this dissertation work was the expansion of FEBio's modeling capabilities in specific areas that are relevant for computational modeling of joint mechanics, although most of the results can be applied to other areas as well. Aside from the new developments discussed in the previous chapters, the author of this work has also contributed to numerous other projects related to FEBio's development. This chapter briefly summarizes some of the most important projects.

8.2 PostView and PreView

FEBio is a command-line application. It takes an input file that describes the finite element model, then solves this model, and finally generates an output file that contains the results of the analysis. This approach was taken because the initial goal was to run FEBio on high-performance machines. This was greatly simplified by avoiding the development of a graphical user interface (GUI). As a result, users needed to prepare the input file and analyze the output file with other software. To facilitate this process, the author developed a graphical finite-element preprocessor PreView and a postprocessor PostView. These programs offer a GUI that uses OpenGL to generate 3D renderings of the model that the user can interact with directly. The preprocessor PreView offers tools for importing meshes from various file formats, generating tetrahedral meshes from surface meshes, applying boundary conditions and materials, and defining the analysis parameters. It also generates the FEBio input file, which can then be passed to FEBio. Alternatively, since the input file is xml-formatted, additional changes to the file can be edited manually if necessary. FEBio can write the results of the analysis to a text file, but in general, the results are stored to a custom binary file format that was designed specifically for FEBio. This file, which is often referred to as the plot file, can be read by PostView. In PostView, the user can visualize the 3D model, add various types of plots (e.g. contour plots, vector plots, iso-surface plots, etc.) and perform different types of analyses (e.g. inspection of values at nodes, facets, and elements, perform statistics, evaluate surface and volume integration, etc.). The user can also filter the data and export it to various data formats for further processing in third party software. All the figures in this work that show finite element models were created with the PostView software.

8.3 FEBio Material Library

FEBio implements an impressive list of constitutive models suited for modeling biological tissues. At the time of writing (FEBio version 2.6), there are 183 materials, many of which were implemented by the author. An exhaustive list of the materials implemented by the author is outside the scope of this work, but a short overview follows.

- *neo-Hookean, Mooney-Rivlin, Ogden*: These Ogden type isotropic hyper-elastic materials are often used to model biological tissues and are also used in FEBio as the basis for several anisotropic material models.
- *trans-iso Mooney-Rivlin*: The transversely isotropic material model by Weiss, Maker and Govindjee is commonly used for modeling fibrous tissues that have a highly aligned fiber distribution. It has been used to model ligament, tendon, and muscle.
- *Muscle, tendon material*: Another material model, which was based on a different set of invariants than is commonly used in hyper-elasticity and was developed by Blemker and Delp, can also be used for modeling muscle and tendon tissue.

- *Veronda-Westmann*: The material by Veronda and Westmann is a phenomenological material model developed for representing isotropic tissues and includes the exponential stiffening that is often observed in fibrous tissues in tension. A transversely isotropic version of this material was also implemented.
- *PRLig*: The author also contributed to the development and implementation of a novel constitutive model for describing tissues with large Poisson's ratios (Swedberg et al.).
- *Visco-elastic*: A visco-elastic material model (Puso and Weiss) was also implemented to describe the time-dependent response of biological tissues. A strength of this formulation is that it can use any of the hyper-elastic material models available in FEBio to define the elastic material response.
- *Damage*: The damage formulation by Simo was combined with the neo-Hookean, Mooney-Rivlin, and trans-iso Mooney Rivlin materials to provide capabilities for modeling damage to biological tissues.

8.4 Material Parameter Optimization

The material parameters used in computational constitutive models are often obtained by fitting model predictions to experimentally obtained results. To accomplish this with FEBio, the author developed a library (called FEBioOpt) that implements several optimization algorithms for estimating material parameters, including the well-known Levenberg-Marquardt method (using the open-source levmar library). Using FEBioOpt, users can define a force-displacement curve, usually obtained from experiment, and find the material parameters that minimize the distance, in a least squares sense, between the experimental curve and the curve predicted by FEBio.

The optimization library has been used by numerous people, but one of the first projects that made extensive use of it was a project that characterized the material properties of the axillary pouch of the glenuhumeral capsule (Rainis et al.). In fact, this project served as the driving application that motivated the development of this library.

8.5 FEBio Contact

The modeling of contact problems has always been very important in many of the applications for which FEBio has been used. For this reason, the author implemented several contact algorithms. These formulations are all based on the augmented Lagrangian approach (Laursen and Maker) and include several tied interfaces, and interfaces for frictionless sliding, sliding with friction, and periodic surface constraints.

The contact modeling framework was later greatly expanded by Gerard Ateshian who added several original contact formulations, including modeling of contact between biphasic materials (Ateshian, Maas and Weiss), and modeling of solute transport across the contact interface (Ateshian, Maas and Weiss). The author contributed to this work by making the necessary modifications to the framework to accommodate these novel formulations, such as contact detection algorithms that can work with all the various element formulations supported by FEBio. The author also implemented support for these novel formulations in PreView.

8.6 Computational Homogenization

Recently, numerical methods have been proposed for including the detailed micro-structure of the material directly in the computational model. Explicit representation of the microstructure into a single a finite element model is still outside the capabilities of modern computer hardware. Instead, researchers have proposed homogenization methods that in a sense average the response of the microstructure. This averaged response is then used to drive a macroscopic model. Several homogenization methods exist, but one approach of particular interest to the FEBio project is a nested finite element computational homogenization method (Kouznetsova). In this method, the microstructure, as well as the macroscopic model are both represented by a finite element model. The macro-model is solved as usual, except during the evaluation of stresses and tangents. Instead of using a traditional constitutive model, the local deformation gradient is passed on to the micro-model. Suitable boundary conditions are defined for the micro-model and then this micro-model is solved as a standard finite element model. Finally, an averaged stress and tangent are evaluated and passed back to the macro-model.

The author implemented a first-order homogenization approach in FEBio (Kouznetsova) which relies on classical continuum mechanics and separation of scales between the macro- and micro-model. The implementation uses FEBio for solving both the macro- and micro-model. The link between the two scales is accomplished via a specialized material model. Currently, the author is implementing a second-order homogenization method, which is based on gradient-elasticity theory (Kouznetsova, Geers and Brekelmans; Nguyen, Becker and Noels). As opposed to the first-order approach, which essentially assumes an infinitesimally small micro-model, this method can model the effects of size of the micro-model. This second-order method requires C1-continuity. To avoid the use of C1 continuous shape functions in FEBio, a discontinuous Galerkin approach is taken that approximates continuity of the shape functions in a weak sense (Nguyen, Becker and Noels). With these homogenization methods, users will be able to include the explicit microstructure into their models and create a more realistic representation of the tissues being modeled.

8.7 Multiphasic Framework

Due to the high water content of many biological tissues, the accurate modeling of the fluid response is very important for many applications. Therefore, a coupled fluid-solver, based on multiphasic theory, has been implemented in FEBio. The author developed, with assistance of Gerard Ateshian, the original “poro-elastic” formulation, which has now been superseded by a more general and more powerful multiphasic modeling framework (Ateshian, Maas and Weiss). This framework now includes the ability to model multiple solutes and chemical reactions (Ateshian et al.). The framework was predominantly developed by Gerard Ateshian, but the author contributed by making the necessary changes to the FEBio code to accommodate this novel framework. This included adding the ability to model an arbitrary number of degrees of freedom and improving the material framework for representing the complex material representation of multiphasic materials. The author also added, with assistance of Gerard Atheshian, support for multiphasic modeling in PreView.

8.8 References

Ateshian, G. A., Steve A. Maas, and J. A. Weiss. "Finite Element Algorithm for Frictionless Contact of Porous Permeable Media under Finite Deformation and Sliding." *J. Biomech. Eng.* 132.6 (2010).

---. "Multiphasic Finite Element Framework for Modeling Hydrated Mixtures with Multiple Neutral and Charged Solutes." *J. Biomech. Eng.* 135.11 (2013).

---. "Solute Transport across a Contact Interface in Deformable Porous Media." *J. Biomech.* 45.6 (2012): 1023-27.

Ateshian, G. A., et al. "Computational Modeling of Chemical Reactions and Interstitial Growth and Remodeling Involving Charged Solutes and Solid-Bound Molecules." *Biomech. Model. Mechanobiol* 13.5 (2014): 1105-20.

Blemker, S.S., and S.L. Delp. "Three-Dimensional Representation of Complex Muscle Architectures and Geometries." *Ann. Biomed. Eng.* 33.5 (2005): 661-73.

Kouznetsova, V.G. "Computational Homogenization for the Multi-Scale Analysis of Multi-Phase Materials." Technische Universiteit Eindhoeven, 2002. Print.

Kouznetsova, V.G., M.G.D. Geers, and W.A.M. Brekelmans. "Multi-Scale Second-Order

Computational Homogenization of Multi-Phase Materials: A Nested Finite Element Solution Strategy." *Comput. Methods Appl. Mech. Eng.* 193 (2004): 5525-50.

Laursen, T. A., and B. N. Maker. "Augmented Lagrangian Quasi-Newton Solver for Constrained Nonlinear Finite Element Applications." *Intl. J. Num. Meth. Eng.* 38.21 (1995): 3571-90.

Nguyen, V.-D., G. Becker, and L. Noels. "Multiscale Computational Homogenization Methods with a Gradient Enhanced Scheme Based on the Discontinuous Galerkin Formulation." *Comput. Method. Appl. M.* 260 (2013): 63-77.

Puso, M., and J. A. Weiss. "Finite Element Implementation of Anisotropic Quasi-Linear Viscoelasticity Using a Discrete Spectrum Approximation." *J. Biomech. Eng.* 120.1 (1998): 62-70.

Rainis, E.J., et al. "Material Properties of the Axillary Pouch of the Glenohumeral Capsule: Is Isotropic Material Symmetry Appropriate?" *J. Biomech. Eng.* 131.3 (2009): 1007-10.

Simo, J.C. "On a Fully Three-Dimensional Finite-Strain Viscoelastic Damage Model: Formulation and Computational Aspects." *Computer Methods in Applications of Mechanics and Engineering* 60.2 (1987): 153-73.

Swedberg, A.M., et al. "Continuum Description of the Poisson's Ratio of Ligament and Tendon under Finite Deformation." *J. Biomech.* 47.12 (2014): 3201-09.

Veronda, D.R., and R.A. Westmann. "Mechanical Charactization of Skin - Finite Deformations." *J. Biomech.* 3 (1970): 111-24.

Weiss, J.A., B.N. Maker, and S. Govindjee. "Finite Element Implementation of Incompressible, Transversely Isotropic Hyperelasticity." *Comput. Method. Appl. M.* 135 (1996): 107-28.