



ANNUAL  
REVIEWS **Further**

Click [here](#) to view this article's online features:

- Download figures as PPT slides
- Navigate linked references
- Download citations
- Explore related articles
- Search keywords

# FEBio: History and Advances

Steve A. Maas,<sup>1</sup> Gerard A. Ateshian,<sup>2</sup>  
and Jeffrey A. Weiss<sup>1,3</sup>

<sup>1</sup>Department of Bioengineering and Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, Utah 84112; email: [jeff.weiss@utah.edu](mailto:jeff.weiss@utah.edu)

<sup>2</sup>Department of Mechanical Engineering and Department of Biomedical Engineering, Columbia University, New York, New York 10027

<sup>3</sup>Department of Orthopedics, University of Utah, Salt Lake City, Utah 84112

Annu. Rev. Biomed. Eng. 2017. 19:279–99

The *Annual Review of Biomedical Engineering* is online at [bioeng.annualreviews.org](http://bioeng.annualreviews.org)

<https://doi.org/10.1146/annurev-bioeng-071516-044738>

Copyright © 2017 by Annual Reviews.  
All rights reserved

## Keywords

FEBio, finite element method, biomechanics, biophysics

## Abstract

The principal goal of the FEBio project is to provide an advanced finite element tool for the biomechanics and biophysics communities that allows researchers to model mechanics, transport, and electrokinetic phenomena for biological systems accurately and efficiently. In addition, because FEBio is geared toward the research community, the code is designed such that new features can be added easily, thus making it an ideal tool for testing novel computational methods. Finally, because the success of a code is determined by its user base, integral goals of the FEBio project have been to offer support and outreach to our community; to provide mechanisms for dissemination of results, models, and data; and to encourage interaction between users. This review presents the history of the FEBio project, from its initial developments through its current funding period. We also present a glimpse into the future of FEBio.

## Contents

1. INTRODUCTION .....	280
2. INITIAL DEVELOPMENT (2005–2009) .....	281
2.1. The FE Kernel .....	281
2.2. PreView and PostView .....	283
2.3. FEBio 1.0 .....	283
3. FIRST FUNDING PERIOD (2008–2012) .....	284
3.1. Multiphasic Framework .....	284
3.2. Enhanced Tetrahedral Elements .....	286
3.3. Pre- and Postprocessor Enhancements .....	286
3.4. Validation and Verification .....	286
3.5. Expansion of Support Mechanisms .....	286
4. SECOND FUNDING PERIOD (2012–2016) .....	287
4.1. Multiphasic Models and Chemical Reactions .....	287
4.2. Plug-Ins .....	288
4.3. Parallelization .....	289
4.4. Quadratic Tetrahedral Elements .....	290
5. IMPACT OF FEBio .....	291
5.1. Overall Impact .....	291
5.2. Specific Use Cases .....	292
6. FUTURE OF FEBio .....	292
6.1. FEBio, a Multiphysics Solver .....	292
6.2. Center for Computational Biomechanics and Biophysics .....	294

## 1. INTRODUCTION

The finite element (FE) method has proven to be one of the most versatile numerical methods for solving computational problems in physics. Although it was initially applied to problems in mechanics, since its inception in the 1950s it has been used in virtually every field of engineering and physics, including solid mechanics, fluid flow, heat transfer, electromagnetism, diffusion, and many more. In the field of biomechanics, the FE method was applied to problems as early as the 1970s (e.g., 1–6). Since that time, the application of FE analysis in biomechanics research and design has increased exponentially as commercial software and improved computing platforms have made the method more accessible to research communities. Applications have ranged from the molecular level to the cellular, tissue, and organ levels.

The field of biomechanics has made tremendous advances over the last few decades as our understanding of the complex structure of biological tissues has grown. As an example, the development of mixture theory as an accurate approach for modeling the interaction of the solid and fluid components of living tissues has resulted in many new insights (7–13). However, 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. The commercial packages that are most commonly used by the biomechanics community are not specifically geared toward biological applications. In addition, the closed nature of commercial codes makes it difficult to verify them (14, 15) or to compare results across different codes because the implementation details are often unavailable. Commercial codes often do not offer a mechanism for adding new features easily, making

it challenging for researchers to implement and test new computational methods or constitutive models.

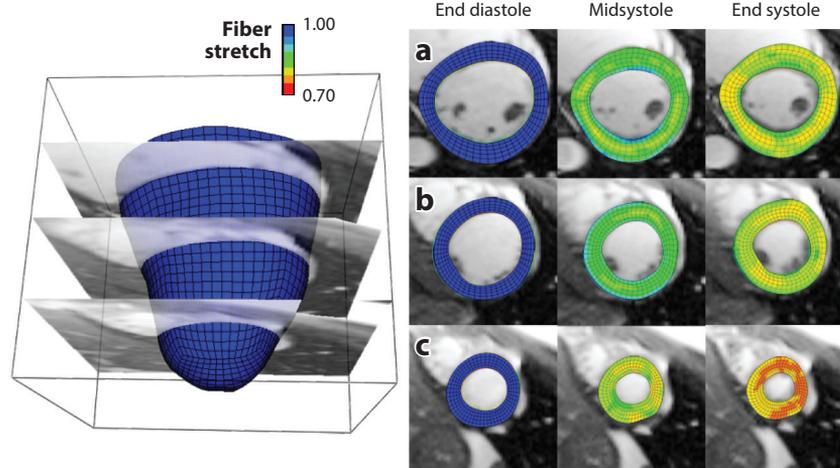
The limitations of commercial codes forced many researchers to develop custom FE packages for solving problems in the biomechanics domain. One of the earliest examples of an FE code developed for applications in biomechanics was that by Simon et al. (16), who developed a custom code for modeling intervertebral disc segments based on porous media theories. FE implementations of biphasic theory for cartilage were presented by Spilker and colleagues for small strains (17) and finite deformation of the solid matrix (18), using penalty methods to enforce incompressibility. They considered various refinements to their approach (19), including hybrid, mixed-penalty, and velocity–pressure formulations (20, 21), as well as tissue anisotropy (22) and explicit modeling of interstitial fluid viscosity (23). u–p (displacement–pressure) formulations of the biphasic mixture theory were presented by Oomens et al. (13) and Wayne et al. (24), and a biphasic formulation accommodating solid matrix viscoelasticity was developed by Ehlers & Markert (25). Custom-written FE codes for modeling deformation, electrokinetics, transport, and swelling of charged hydrated tissues were developed by Simon and colleagues (26, 27), Levenston et al. (28), Sun et al. (29), and Frijns et al. (30). 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 rather were 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 (around 2005) none of these custom-written codes offered extended documentation, support, or mechanisms for continued development.

Due to the many limitations of commercial and custom-written FE packages, the authors of this review (and many others in the research communities) felt there was 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 focusing on features that are relevant in the field. These would include, for example, 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 freely available and designed such that it would be easy for researchers to implement new algorithms (e.g., new constitutive models). This would greatly reduce the need for new custom-written codes and overcome the problems with verification and repeatability of results as researchers can easily share their new additions and algorithms. Third, emphasis would be placed on thorough documentation, support, and outreach to the community, which would make it much easier for researchers to develop new ideas and share them with others.

## 2. INITIAL DEVELOPMENT (2005–2009)

### 2.1. The FE Kernel

In 1995, Weiss and colleagues were involved in the development of a deformable image registration algorithm termed Hyperelastic Warping (31). This method uses an FE method to solve the image registration problem. The result of this approach is a deformation that is physically realistic and can be used to calculate the strains and, assuming the constitutive properties of the materials are known, the stresses as well. This algorithm was originally implemented in NIKE3D,



**Figure 1**

(*Left*) Example of an application of the Hyperelastic Warping method, which combines a finite element kernel with an image registration technique. (*Right*) Left ventricular fiber stretch developed through systole (left column, end diastole, fiber stretch = 1.0; middle column, midsystole; right column, end systole) with the Hyperelastic Warping solution depicted in (a) basal, (b) midcavity, and (c) apical slices.

a nonlinear implicit FE code for solving problems in solid mechanics, developed at Lawrence Livermore National Laboratory (LLNL). Weiss and colleagues (32–34) obtained the source code for NIKE3D through their LLNL licensing mechanism and used the code to solve problems in computational biomechanics. A special version of NIKE3D was developed that included the Hyperelastic Warping algorithm. Unfortunately, NIKE3D was not free, and as a government-owned program, the dissemination of the additions was complicated. Consequently, the algorithm was not widely used outside Weiss’s lab despite the fact that it had many valuable applications (**Figure 1**) (35–37). To overcome this problem, Weiss wanted to create a new, custom, open-source FE kernel that would drive future developments of the warping algorithm.

The development of this FE kernel commenced in 2005, at the same time that Maas joined Weiss’s lab. It was quickly realized that the kernel could be used to solve traditional FE problems as well, and in fact could replace the dependency on NIKE3D. This became a pressing concern because LLNL reported that it would soon discontinue development of NIKE3D. This kernel code was greatly expanded in 2005 and 2006 and became the predecessor to FEBio. When other researchers expressed interest in our new code, the idea of FEBio was born: a new FE code specifically designed for the biomechanics community. Ateshian, who had previously collaborated with Spilker et al. (38) and later developed his own custom FE codes to analyze contact between biphasic cartilage layers (39, 40), became one of the earliest researchers to express interest in the FEBio project. With his expertise in mixture theory, Ateshian joined forces with Weiss and started working on the incorporation of biphasic modeling capabilities in FEBio, a capability that was of significant interest to the biomechanics community.

The main focus of the first version of FEBio was the implementation of a nonlinear, implicit FE solver for large deformation mechanics. The formulation was based on the discretization and linearization of the virtual work equation (41) and the implementation of robust quasi-Newton methods for solving the nonlinear FE equations (42). Constitutive models were mostly defined using hyperelastic strain energy density functions. Nearly incompressible materials were implemented using uncoupled formulations and a three-field variational formulation to overcome

locking issues (43). Transverse isotropy was also supported in some of the material models (44), as was viscoelasticity (45). The implementation of viscoelasticity allowed any of the hyperelastic materials to be used as the elastic component. A special material type was defined for rigid materials, which implicitly defined the rigid parts of the model. Rigid bodies were modeled using an efficient method based on the reduction of degrees of freedom (46). Frictionless sliding and tied contact algorithms were implemented using an augmented Lagrangian approach (47), although a standard penalty formulation was available as well. A biphasic material was also supported, which required the specification of the elastic mixture response material and the permeability of the solid phase. When such a material was defined, FEBio would essentially solve a coupled displacement–fluid pressure problem in which the virtual work equation combines the mixture momentum and mass balance equations through a monolithic approach (40).

## 2.2. PreView and PostView

FEBio was developed as a command-line tool. This choice was motivated mostly by our initial intent to continue using our available pre- and postprocessing software and focus our efforts solely on the development of the FE solver. Thus, FEBio would take an input file, solve the problem, and return the results in one or more output files. We took advantage of our experience with NIKE3D and initially used the same input and output file formats. This allowed us to continue using the commercial code TrueGrid (<http://truegrid.com>) to generate the input file and LSDYNA PrePost (<http://lstc.com>) for postprocessing. Of course, this presented a problem for other users of our code because they would be required to purchase TrueGrid and LSDYNA in order to use FEBio. We reasoned that it did not make sense to offer a free software package that required a significant financial investment in order to use it, motivating us to develop our own pre- and postprocessing software.

Another limitation was that these initial file formats were static. The structure and contents of the files were predefined and could not be easily changed. Because one of the aims of the FEBio project was to allow researchers to add new features, this limitation would severely complicate the process of adding new input data and storing new output data fields. Therefore, we designed a new, extendable input format (the “feb” format) and output format (the “xplt” format). Being able to support these new formats was another motivation for the development of custom pre- and postprocessing software.

The FE preprocessor, termed PreView, initially allowed the import of volume meshes and the definition of boundary conditions, material, and analysis parameters. Users then exported the model definition as an FEBio input file. This initial implementation still posed the problem of how to generate the FE meshes. However, at that time, it was felt that many software packages, both commercial and free, were available to achieve these tasks, so this issue was not considered a severe limitation.

PostView was developed for postprocessing and analysis of the FEBio results. The initial implementation of PostView was actually an adaptation of another software package, called WarpLab, which Maas had developed to analyze the results of the Hyperelastic Warping algorithm.

## 2.3. FEBio 1.0

The first version of FEBio was released in December 2007. It was developed using the C++ programming language. This was a break from tradition because many FE codes at the time were still being developed in FORTRAN. In fact, most of our initial experience with FE codes was in FORTRAN due to our work with NIKE3D, which was written in FORTRAN. However, at

the time C++ had become an accepted language for scientific programming, mostly as a result of its maturation as a language and the availability of efficient compilers that made it competitive with FORTRAN (48). The greatest advantage of C++ over FORTRAN was its support for object-oriented design, which greatly simplified the development of large, complex software. This decision was vindicated after initial benchmark problems showed that FEBio outperformed NIKE3D quite well. Of course, this improved performance cannot be attributed solely to the chosen programming language, as NIKE3D was optimized for the now mostly obsolete vector-processing machines, whereas FEBio was designed to target more modern computer architectures. Nevertheless, the excellent initial performance of FEBio was encouraging.

As part of the first FEBio release, a forum was created where users could submit questions, report bugs, or propose new features. The software was made available for download from the website of Weiss's Musculoskeletal Research Laboratory (MRL) at the University of Utah. Both prebuild binaries and the source code itself were available for download.

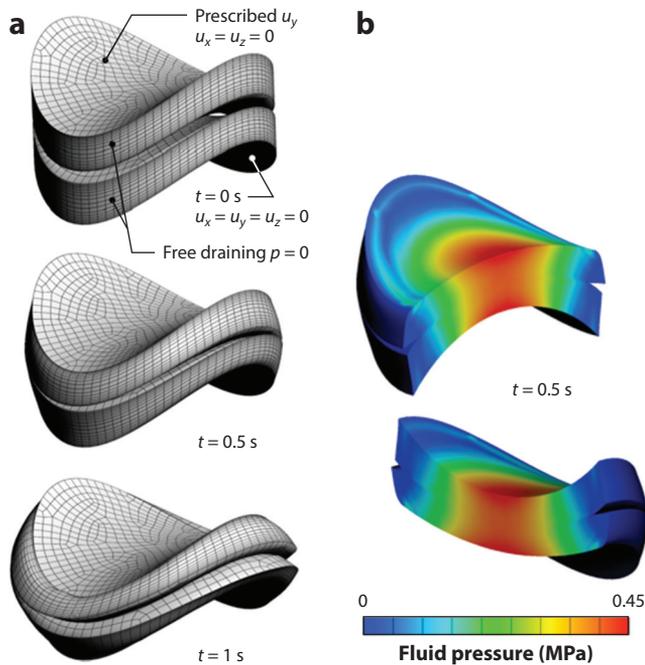
### 3. FIRST FUNDING PERIOD (2008–2012)

#### 3.1. Multiphasic Framework

In 2008, we received our first federally funded grant for the FEBio project through the US National Institutes of Health. The main goal for this funding period was to expand FEBio's capabilities for modeling problems in biomechanics. In particular, the initial implementation of biphasic (or poroelastic) materials was to be extended to mixtures with multiple constituents, using the general framework of mixture theory. This allowed the representation of mixtures consisting of a solid matrix, a solvent, and initially only one (40) but later any number of solutes. A special emphasis was placed on solute–solid matrix interactions, including solute exclusion from a fraction of the matrix pore space (solubility) and frictional momentum exchange that produces solute hindrance and pumping under certain dynamic loading conditions (49). The FE formulation implemented full coupling of mechanical and chemical effects, providing a framework in which material properties and response functions may depend on solid matrix strain as well as solute concentration. The implementation was verified using canonical problems for which analytical or alternative numerical solutions exist. Additional boundary conditions were also implemented to accommodate the complete set of boundary conditions applicable to a solid–fluid mixture, including prescribed solid displacement, fluid pressure, total traction, effective traction, normal fluid flux, and fluid volumetric flow rate. A general theoretical framework for formulating constitutive relations for anisotropic, strain-dependent permeability and solute diffusivity in multiphasic mixtures was also developed, as some of the fundamental principles for such formulations were lacking in the prior literature (50).

Furthermore, to accommodate contact between solids and mixtures having a solid matrix, new contact algorithms were implemented that properly accounted for conservation of mass and momentum between contacting mixtures. These contact algorithms allowed both the solvent (**Figure 2**) and the solutes (**Figure 3**) to flow across the contact interface (51, 52).

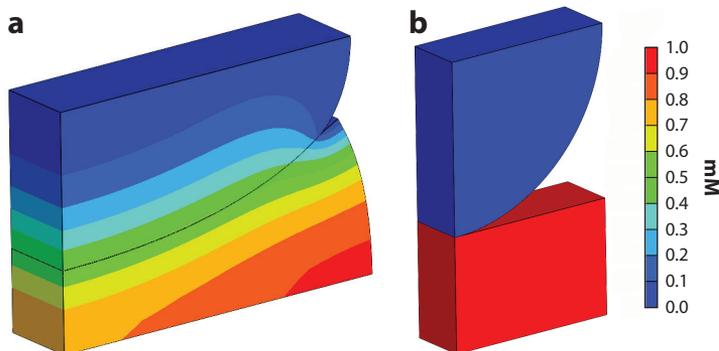
A particularly useful feature of FEBio has been its ability to model mixtures of solid constituents, allowing users to combine various models available within the solid material library to produce custom materials with very little effort. This feature has been particularly useful for modeling fibrous tissues as mixtures of one or more fiber families that only sustain tension, superposed on a ground matrix that may also support compression. Without a ground matrix, such tension-only material models would otherwise be unstable (53). The ability to model biological tissues using continuous fiber distributions has also been added to FEBio, as these models produce far



**Figure 2**

Biphase contact between saddle-shaped articular layers. (a) Boundary conditions and mesh deformation at various time points ( $t$ ). (b) Fluid pressure distribution shown on cross sections at  $t = 0.5$  s. Note the continuity of the fluid pressure across the contact interface.

better agreement between theory and experiments for some tissues (54). A special class of solid mixtures has been introduced, allowing the various solid constituents to have distinct reference configurations; this feature makes it possible to model multigenerational growth of biological tissues, including the evolution of residual stresses with evolving growth (55).



**Figure 3**

Contact analysis with solute transport. (b) Initial configuration. The solute concentration is initially 1 mM in the bottom body and 0 mM in the top body. (a) During finite deformation contact, solute transports from the bottom body to the top body, enforcing concentration continuity at the contact interface.

### 3.2. Enhanced Tetrahedral Elements

Another important goal of this funding period was the implementation of tetrahedral formulations that are suitable for large deformation mechanics, especially for modeling near-incompressible materials. As is well known, linear tetrahedral formulations perform poorly under these conditions and have a tendency to lock; in other words, they grossly underpredict the displacements under applied loads (56). Yet the use of linear tetrahedral elements is appealing in FE modeling due to the ease with which complex geometries can be meshed with such elements. To overcome the limitations of linear tetrahedral elements, an enhanced tetrahedral element formulation (the UT4 element) was implemented in FEBio (57, 58). This formulation evaluates the FE equations at the nodes as the average of the quantities evaluated at the adjacent elements. However, the nodally averaged results are inherently unstable, so a weighted sum of the standard linear tetrahedral residual and the new nodally averaged one is used to improve stability. Unfortunately, despite some promising initial results for certain test problems, this element formulation ultimately demonstrated inaccurate results, especially for contact problems, and spurious deformation modes (59).

### 3.3. Pre- and Postprocessor Enhancements

During this funding period, our existing pre- and postprocessing software packages were also expanded. Support for tetrahedral mesh generation was added using TetGen (<http://wias-berlin.de/software/tetgen/>), which allowed users to convert triangulated surfaces (created, for instance, with image segmentation software) into tetrahedral FE meshes. Support for boundary conditions on multiphase mixtures was also implemented, allowing users to prescribe fluid pressures, fluid flux, solute concentration, solute flux, and mixture normal traction as well as to apply contact interfaces that take fluid and solute transfer between the contacting surfaces into account. Finally, support for defining biphasic, triphasic, and multiphase materials was added in PreView.

### 3.4. Validation and Verification

For any simulation software to gain credibility, it must be subjected to a well-defined verification and validation process. FEBio was verified by comparing its predictions with both analytical solutions and results obtained with other FE codes (60). In addition, a rigorous regression test suite was developed that ran nightly and ensured that FEBio produced consistent results. The test suite currently contains more than 160 problems. The test suite was also made available to users so that they could ensure that FEBio worked properly on their computing platforms as well.

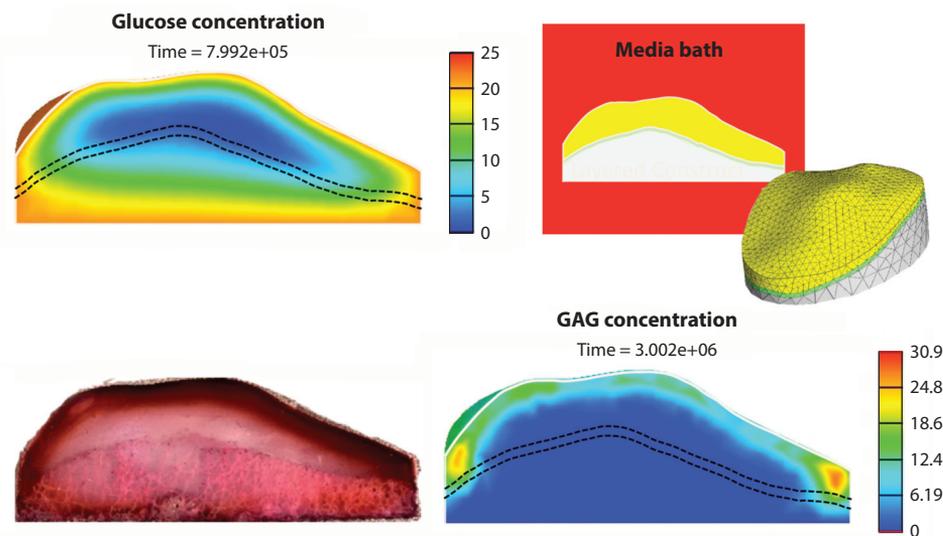
### 3.5. Expansion of Support Mechanisms

An integral part of the FEBio project was providing mechanisms for user and developer support, as well as encouraging interactions between members of our community. The MRL website (<http://mrl.sci.utah.edu>) was completely redone and launched in February 2010. Registered users could download FEBio, PreView, and PostView executables for Windows, Mac, and Linux operating systems. The source code for FEBio was also made available for download. Manuals were made available online for the first time, along with downloadable PDF files, and an updated help system with search capabilities was implemented. The online FEBio source code documentation was generated with Doxygen (<http://www.doxygen.org>).

## 4. SECOND FUNDING PERIOD (2012–2016)

### 4.1. Multiphasic Models and Chemical Reactions

During the second funding period, the multiphasic framework was expanded to include modeling of any number of neutral or charged solutes in a porous deformable solid matrix that may carry electrical charge (61). The inclusion of solutes and a charged solid matrix made it possible to model osmotic effects that play a dominant role at the cellular level (62) and in a variety of soft tissues that exhibit a significant fixed charge density (63–65). Thus, it became possible to analyze Donnan osmotic pressure, electro-osmosis, electrodiffusion, and other electrokinetic phenomena that are relevant to biological tissues. Chemical reactions were also incorporated into the code to allow mass exchanges between solutes and solid-bound molecular species. Any number of chemical reactions could be included in a particular multiphasic mixture material. Four standard constitutive relations were incorporated into the code; these included the law of mass action for forward reactions, the law of mass action for reversible reactions, Michaelis–Menten kinetics as a special case of the law of mass action, and the solid remodeling theory of Huiskes and colleagues (66). The validity of the code was verified for a number of problems for which analytical solutions were available (**Figure 4**) (67). For example, these implementations made it possible to investigate the role of nutrient transport and consumption in tissue engineering studies (68, 69) as well as



**Figure 4**

A finite element model of a human patella-sized engineered tissue construct (*mesh shown on the right*) was analyzed using chemical reactions to model glucose consumption from the available media bath (*top right*), as well as glycosaminoglycan (GAG) synthesis in response to glucose availability. The glucose concentration at an intermediate time point during the growth process (*top left*) indicates limited glucose availability near the center of the construct due to consumption by cells at the construct's periphery. The concentration of synthesized GAGs at the final time point in simulated culture (*bottom right*) demonstrates that GAG synthesis occurred predominantly near the articular surface. The Safranin-O staining of a real tissue construct, cultured under the conditions simulated in the finite element model, is shown in the bottom left panel. Agreement between the model and experimental results validates the modeling method.

the binding kinetics of growth factors to the scaffold, extracellular matrix, and cells in a tissue construct (70).

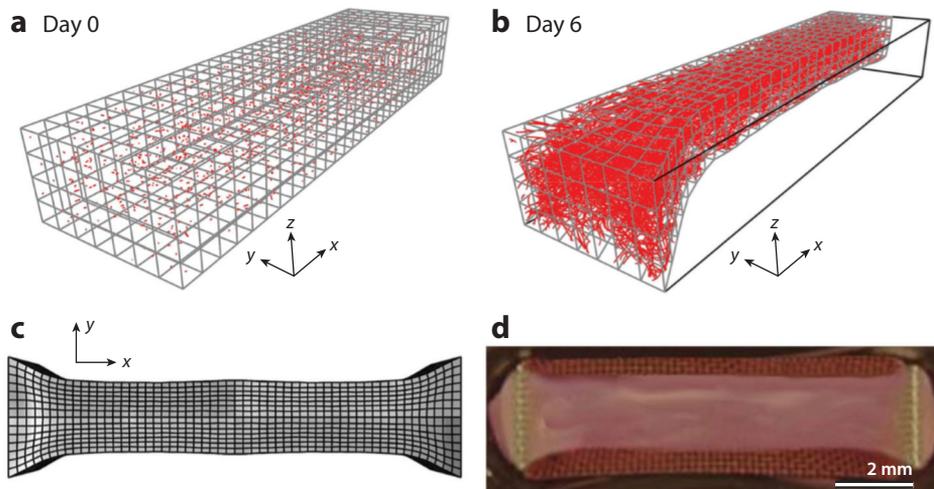
In a development related to the expansion of the multiphasic framework, the implementation of body forces in biphasic materials served as a template for the implementation of active solute transport in the form of a solute momentum supply. Ongoing studies aim to demonstrate the application of this reactive framework to model active transport processes in cells, such as those involved in cell volume regulation, pH regulation, and calcium signaling.

Our formulation of a reactive mixture framework for constrained solid mixtures also produced unexpected benefits, such as the ability to reproduce the classical theory of linear viscoelasticity by using bond breaking and reforming reactions (rather than the classical springs and dashpots); with this novel approach to a classical problem, we were able to formulate more general theories of nonlinear viscoelasticity (71), which have now been implemented in FEBio. The same reactive constrained mixture framework has also allowed us to model damage mechanics in tissues by modeling the permanent bond-breaking process as a reaction (72); this damage framework has also been implemented in FEBio, allowing users to implement damage in any of the existing solid materials by using a variety of damage criteria. We have used this model to investigate damage mechanics in cartilage constructs that are subjected to excessive Donnan osmotic swelling due to a more rapid synthesis of charged proteoglycans than a restraining collagen matrix (72).

The FEBio framework has been designed such that users may combine any of these models as needed. For example, a multiphasic material may be set up with a charged solid matrix and multiple neutral or charged solutes; the solid matrix may be modeled as viscoelastic or may be allowed to undergo damage in response to loading, at the same time that growth and degradation processes are taking place, driven by chemical reactions involving solutes and solid-bound molecules. Importantly, the material properties may also evolve with the evolving composition, so that tissue remodeling analyses may be easily accommodated.

## 4.2. Plug-Ins

One of the important design goals for FEBio was to make it easy for users to expand the source code and add new features. The initial versions of FEBio used principles from object-oriented programming to create a modular structure of the code that could be modified easily. Although it was possible for users to download the source code and modify it directly, this process presented some practical limitations. A significant issue was that after the user modified the code, this customized version might not be compatible with the next FEBio upgrade, requiring significant effort to port changes to the new version. Another difficulty for developers was the need to build the entire FEBio source code tree from scratch, even if they needed to make only a small modification. Because FEBio depends on third-party libraries (e.g., MKL for PARDISO linear equation solver, levmar for optimization, zlib for data compression), some of them not free, users would be required to obtain these third-party libraries as well. To overcome this limitation, a plug-in framework was developed for FEBio that allowed users to create customized tools without the need to modify or recompile the FEBio source code. The plug-ins were implemented as stand-alone libraries (dynamically linked libraries on Windows or shared object libraries on Linux and Mac) that are linked to the FEBio executable at run time. Each plug-in implements a new feature by extending the classes that are defined in the FECore library, which contains all the base classes that are used by FEBio. The plug-ins are compiled and built separately from the FEBio code. To link the plug-in with FEBio, the user simply adds the path to the plug-in to the FEBio configuration file, and FEBio automatically loads the plug-in at run time and adds the classes defined in the plug-in to the FEBio framework. This plug-in feature allows users to extend almost every part of FEBio;



**Figure 5**

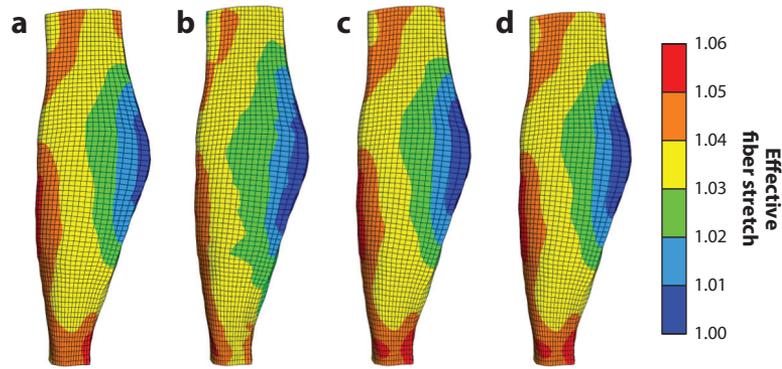
Example illustrating the application of the FEBio angiogenesis plug-in. Simulation of a long axis constrained (LAC) vascularized gel. (a) At day 0, the mesh was seeded with initial microvessel fragments. (b) At day 6, the gel was highly perfused by microvessels and deformed into a “neck” shape, as observed in the LAC experiments. (c) A full-geometry construction of the gel as predicted by the deformed finite element mesh. (d) The deformed gel geometry predicted in the simulation closely resembled the shape of LAC-vascularized gels at day 6.

for instance, they may add a new material model, customize or extend the plot file output, couple FEBio to other codes, and much more.

For example, a plug-in was developed to couple FEBio to a computational model of angiogenesis (73). The coupling allowed us to include matrix deformations and study the effects thereof on the growth of the microvessel fragments (**Figure 5**). Another plug-in implemented Weiss and colleagues’ (31) original Hyperelastic Warping algorithm, which now makes use of FEBio as its FE kernel. This plug-in adds image-based body forces that register an FE mesh onto a target image (**Figure 1**). A plug-in was also developed for applying prestrain to models (74). This plug-in allows users to apply an initial strain to the reference configuration (**Figure 6**), an important application for modeling biological tissues, or to prestress a geometry by solving an inverse FE problem. These three examples illustrate the versatility of the plug-in framework and demonstrate that it provides a powerful tool for extending the capabilities of FEBio.

### 4.3. Parallelization

Initially, FEBio did not take advantage of parallel architectures on desktop computers, at least not directly. Support for parallelization was a result of using third-party parallel linear solver libraries (e.g., Intel MKL PARDISO; <http://software.intel.com/en-us/articles/intel-mkl-pardiso>). The justification for this design choice was that, as an implicit FE solver, the factorization of the global stiffness matrix takes up most of the solution time. However, the use of quasi-Newton methods (42), which do not require the factorization of the stiffness matrix at each iteration, meant that more time was spent in the serial part of the code. In addition, the development of more complex constitutive models [e.g., continuous fiber distribution materials (75)] significantly increased the time spent evaluating the stiffness matrix and right-hand-side vector. The result was that the



**Figure 6**

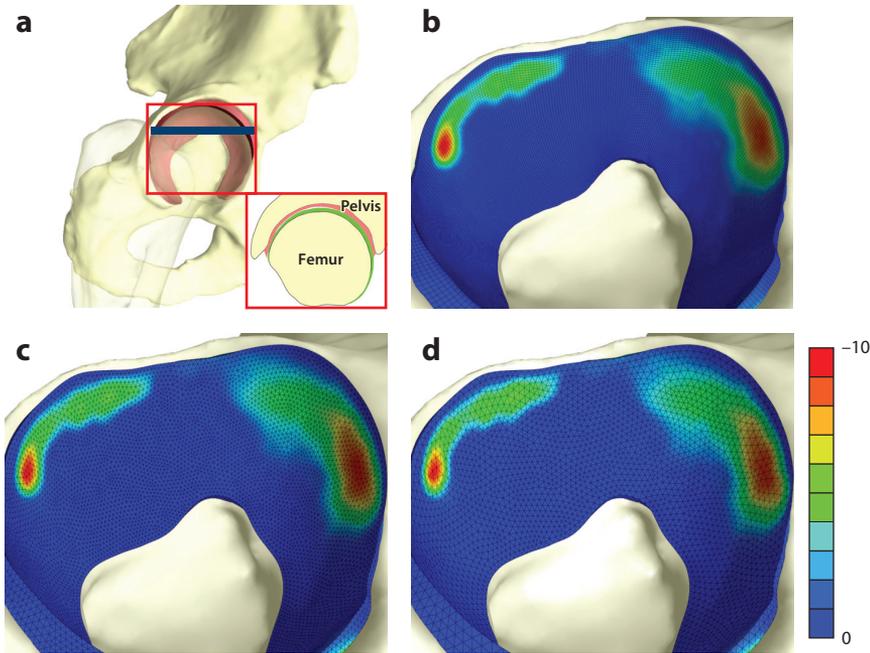
Effects of applying in situ stretch to a reference geometry that is kinematically incompatible. (a) Reference geometry with target in situ fiber stretch. (b) Deformed state and effective fiber stretch without updates. (c) Effective fiber stretch after eliminating distortion. (d) Effective fiber stretch after enforcing the target in situ stretch.

serial part of the code had now become a significant bottleneck; to overcome this limitation, the serial parts of the code had to be parallelized. OpenMP (<http://openmp.org>) was chosen as the parallelization mechanism because it was the easiest way to parallelize a serial code. The parts of FEBio that were most time consuming were identified and, where possible, were rewritten with support for OpenMP. These included most stiffness matrix calculations and right-hand-side evaluations where the loop over all elements was parallelized. Some aspects of the contact projection algorithms, which turned out to be costly in terms of run time, were also parallelized. Some of the error handling routines also had to be rewritten because they were not thread-safe. This initial effort resulted in some encouraging benchmarks. The initial parallelization effort left the assembly of the global structures from the element structures as a critical section, so that this part of the solution process remained serial. This limitation was later removed and resulted in slightly better performance of the parallel code. Additional improvements were achieved through experimentation with different loop splitting mechanisms.

Although the parallel version of FEBio (starting with FEBio 2.0) greatly improved the performance for many problems, the overall performance gain was highly problem dependent and was affected by the time spent inside the linear solver. The most dramatic reductions in run time (up to a factor of 3.5 on architectures with eight processors) were observed for problems that spent the most time outside of the linear solver, such as contact problems or problems that use complex constitutive models.

#### 4.4. Quadratic Tetrahedral Elements

Because the UT4 elements did not prove as successful as initially anticipated, especially for contact problems, the search for efficient tetrahedral formulations continued. This time, the focus was on quadratic tetrahedral formulations. During this grant period, several quadratic tetrahedral element formulations were implemented in FEBio. These elements are an attractive alternative to linear tetrahedral elements because they maintain the advantages of tetrahedral mesh generation. In addition, they can represent curved boundaries more accurately than linear elements because their edges and faces are curved, reducing the need for fine meshes for representing curved structures. Moreover, as reported in a recent publication (76), quadratic tetrahedral elements perform



**Figure 7**

(a) Finite element model of the human hip, including the acetabulum, proximal femur, and articular layers of the acetabulum and femoral head. The red square shows the approximate area corresponding to the contour plots. The blue line shows the approximate location of the inset, depicting a cross section of the model. (b–d) Fringe plots of the third 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 third principal stress and the contact area predicted by the three models.

similarly and sometimes even better, both in terms of computational cost and in terms of accuracy, than the “gold standard” linear hexahedral element (**Figure 7**).

## 5. IMPACT OF FEBio

### 5.1. Overall Impact

The FEBio user community continues to grow. Since the first release of FEBio in 2007, there have been more than 110,000 software downloads. Currently, there are more than 5,100 registered FEBio users and around 1,100 registered users on the forums, and these numbers continue to grow. Our user base ranges from students, some of them new to FE modeling, to more experienced academics and researchers. FEBio is also used as a teaching tool in various institutes across the world, such as the University of Michigan, Boise State University, and Brigham Young University in the United States; Technical University of Madrid in Spain; National University of Singapore; University of Strathclyde in the United Kingdom; Eindhoven Technical University in the Netherlands; Tel Aviv University in Israel; McGill University in Canada; and several others. FEBio has also been featured in many posters and podium presentations during various conferences over the years.

Although these statistics provide a glimpse of the positive impact of FEBio on the community, perhaps the best way of estimating its impact is by looking at how the software has been used by other researchers. Currently, we know of at least 216 articles in peer-reviewed publications that have used FEBio in a significant way, 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. An exhaustive review of these articles is beyond the scope of this article (see <http://febio.org> for the full list). Instead, a few selected examples are presented below.

## 5.2. Specific Use Cases

Few commercial codes offer effective sliding contact algorithms under large deformations for poroelastic materials. ABAQUS (<http://abaqus.com>) has emerged as the de facto gold standard in biomechanics (77–86). It offers a contact algorithm that may be customized with user-specified routines to accommodate fluid pressure and fluid flux interface conditions compatible with biomechanics applications, such as articular joint contact (87). With the introduction of a biphasic contact algorithm in FEBio (51), at least two independent studies aiming to compare ABAQUS and FEBio have recently been published in the biomechanics literature (83, 88). Both studies demonstrated favorable agreement for a variety of test problems, concluding that either code could be used effectively for such applications.

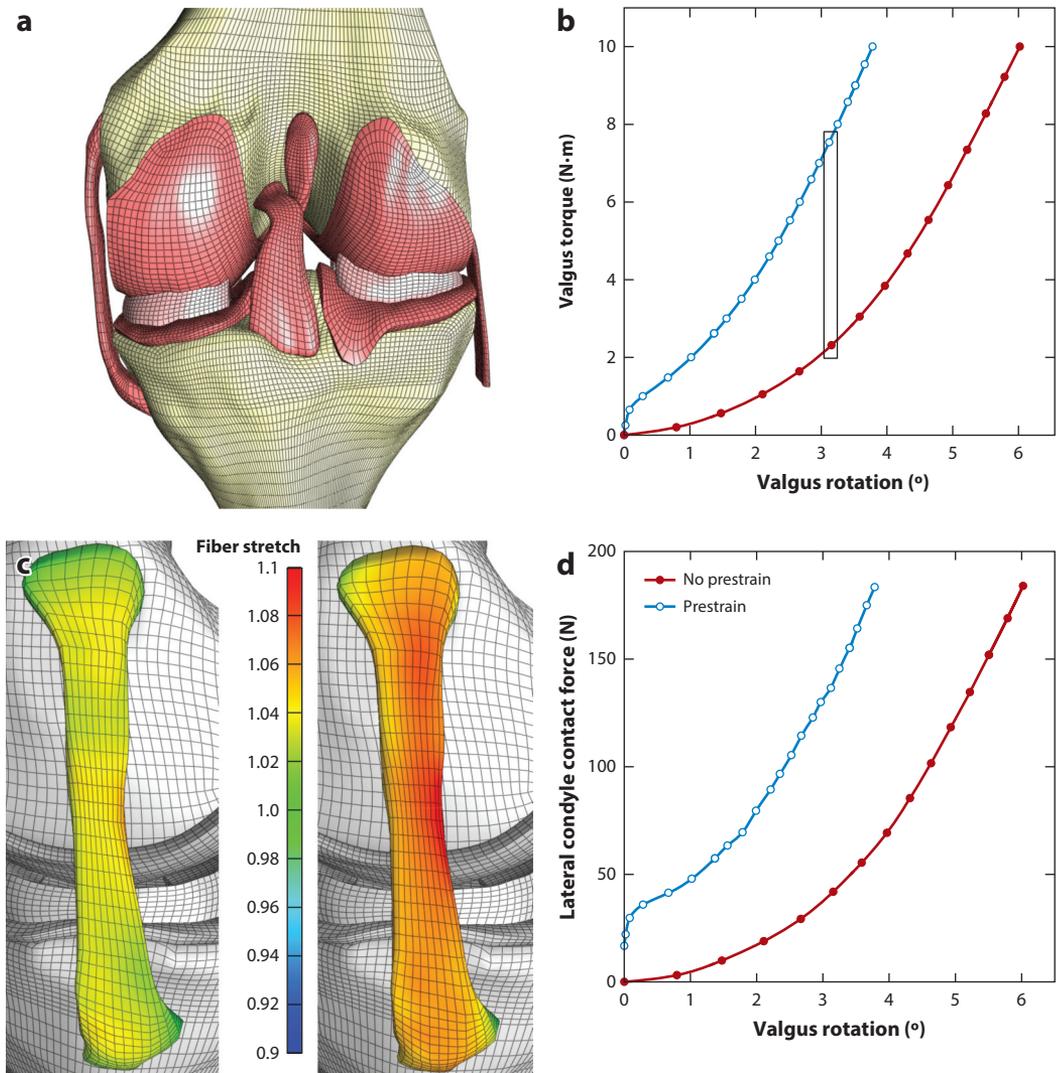
Pierrat et al. (89) implemented a new constitutive model of slightly compressible transversely isotropic material that overcomes some of the limitations of the traditional approach to implementing nearly incompressible formulations. This material model was implemented in both ABAQUS (as a user subroutine) and FEBio (as a material plug-in), and the performance of the codes were compared. Although the results from both codes agreed well, FEBio was notably faster. Due to FEBio's tensor class and operator library, the implementation in FEBio was considerably easier. In addition, FEBio offers a diagnostic routine for checking the consistency between the stress and tangent implementations, providing confidence that the implementation is correct. ABAQUS does not offer such capabilities, making the implementation of a new constitutive model much more difficult.

The OpenKnee project (90, 91) is a collaborative effort between several institutes and research labs; one of its main goals is the development of freely available FE models of the human tibiofemoral joint. These models include detailed anatomically correct models of the femur and tibia, cartilage layers, medial and lateral menisci, and cruciate and collateral ligaments. The models are available for download as FEBio input files. The OpenKnee project showcases how new synergetic collaborations have been formed from the FEBio project and how these new collaborations have driven FEBio development. For example, one of the outcomes of this collaboration was the development of a plug-in that allows users to apply prestrain to FE models, an important application for accurate modeling of biological tissues (Figure 8). The expansion of FEBio's rigid connector features was also motivated in part by this project.

## 6. FUTURE OF FEBio

### 6.1. FEBio, a Multiphysics Solver

FEBio's main priority continues to be the development of a software environment that is tailored to the specific needs of the biomechanics and biophysics communities. The initial focus of the project was mostly on solid mechanics. Following the implementation of biphasic, and later multiphase,



**Figure 8**

Finite element analysis of the OpenKnee model under 10 N·m of valgus torque. (a) The OpenKnee 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 box shows the two points where the fiber strains are compared in panel c. (c) Effective fiber stretch at  $\sim 3.1^\circ$  valgus rotation (*left*) without prestrain and (*right*) with prestrain. (d) The contact reaction force across the lateral condyle. Results demonstrate that the structural behavior of the joint and the predicted ligament strains are dramatically different without a prestrain field to the ligaments; in particular, the response is too soft under valgus torque without prestrain.

capabilities, including the modeling of an arbitrary number of solutes, electrokinetics, and chemical reactions, FEBio has evolved into a true multiphysics code. This allows researchers to model and investigate the interactions between different types of physics within a single model. These capabilities are obviously important for solving problems in biophysics. Due to the importance of modeling multiphysics capabilities, we are currently focusing on expanding FEBio's capabilities in this area.

The modeling of fluids remains one of the biggest challenges faced by FEBio. Although fluids can be modeled in FEBio as components of mixtures, the explicit modeling of viscous fluid flows cannot yet be done. The modeling of computational fluid dynamics, and more significantly the incorporation of fluid–solid interactions (FSIs), is very relevant to biomechanics and biophysics (e.g., cardiovascular mechanics, cerebrospinal mechanics, vocal fold and upper airway mechanics, viscous flow over endothelial cells, canalicular and lacunar flow around osteocytes, diarthrodial joint lubrication, and many applications in biomedical device design). Therefore, the next stage of FEBio development efforts is focusing on the implementation of algorithms that model computational fluid dynamics and FSIs. To accommodate these features, FEBio’s linear and nonlinear solution algorithms will need to be expanded. In particular, the implementation of efficient iterative linear solvers and preconditioners will be essential.

As we expand the capabilities of FEBio into the fluid mechanics and FSI domains, adaptive mesh refinement (AMR; local mesh refinement and coarsening) will become necessary. The accuracy of an FE solution is intimately tied to the discretization because it determines the spatial resolution of both the geometry and the interpolation of field variables. As an example application motivating the need for AMR, high gradients in velocity for fluid mechanics simulations necessitate local refinement for both accuracy and sometimes local convergence as well. These areas are not always possible to identify before the analysis begins, requiring the initial mesh to be refined during the analysis. Although AMR is available in some commercial and free FE packages for solid mechanics, it is much more of a necessity in computational fluid dynamics (CFD) and FSIs, in which the solution times may be very long and restarting an analysis with a new mesh is not practical. FSI algorithms that use arbitrary Lagrangian–Eulerian (ALE) methods give rise to a special necessity for mesh adaptivity at solid–fluid boundaries.

## **6.2. Center for Computational Biomechanics and Biophysics**

The principal investigators of the FEBio project, Weiss and Ateshian, are currently in the process of setting up the Center for Computational Biomechanics and Biophysics (CCBB). The purpose of the CCBB is to continue the development of FEBio and extend it in exciting new directions, as well as to create new collaborations and widen the reach and impact of FEBio.

The integration of image data in FE analyses is another goal, which will require the development of an entire modeling pipeline from image data, through segmentation and mesh generation, to FE analysis. In addition, this new capability would allow users to integrate image data directly into FE models, for instance, for specification of inhomogeneous material properties.

The capabilities of the FEBio preprocessor, PreView, and postprocessor, PostView, will be greatly expanded as well. Users will be able to load image data into PreView and use it for a variety of applications, for instance, to define inhomogeneous material parameters. A powerful geometry engine will be added to PreView (most likely based on the open-source OpenCascade library; see <https://www.opencascade.com/>) that will allow users to combine computer-aided design (CAD)-like geometries with surface geometries from image data and other sources. PostView already has some initial support for loading and visualizing image data, but these capabilities will be expanded greatly. A scripting feature will be added to both PreView and PostView that will allow users to create custom scripts for automating their modeling pipelines. The plug-in capability, which currently exists only in FEBio, will be supported by PreView and PostView as well.

Another development effort will be focused on the expansion of the plug-in framework such that FEBio could be coupled easily with other, domain-specific software. Also, support for scripting languages such as Python will allow users to easily develop new applications with FEBio.

The CCBB center will also focus on expanding our dissemination and training efforts. New education and training programs will be developed for our major target communities, requiring new curricula for course-based education, offering specific tutorials and workshops for relevant audiences, and expanding our already extensive web-based documentation and online education tools.

## DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

## ACKNOWLEDGMENTS

The development of FEBio was supported by the National Institute of General Medical Sciences of the National Institutes of Health under award number R01GM083925. The content of this review is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

The FEBio project has benefited from the help of many contributors. We express our special thanks to David Rawlins, who has been a member of the FEBio software development team since 2008 and has contributed, among other things, to the material optimization module, the distribution of the code and documentation, and the proper functioning of the forum and website. Others have contributed by implementing specific features or offering valuable feedback that helped us address shortcomings that led to code improvements: Dr. Ben Ellis, Dr. Alexander Veress, Dr. Lowell Edgar, Dr. Sam Evans, Dr. Mike Puso, Dr. Ahmet Erdemir, Mr. Robert Nims, and Mr. Chieh Hou. The FEBio project also received a generous donation from Dr. Siswanto from the Universiti Tun Hussein Onn Malaysia. Finally, we express our gratitude to our user community. The level of enthusiasm our users have expressed over the years has greatly exceeded our expectations. We appreciate the invaluable feedback that they have given us, as this has in a profound manner helped us shape FEBio such that it will continue to be the preferred tool for solving computational problems in biomechanics and biophysics.

## LITERATURE CITED

1. Belytschko T, Kulak RF, Schultz AB, Galante JO. 1974. Finite element stress analysis of an intervertebral disc. *J. Biomech.* 7:277–85
2. Davids N, Mani MK. 1974. A finite element analysis of endothelial shear stress for pulsatile blood flow. *Biorheology* 11:137–47
3. Doyle JM, Dobrin PB. 1971. Finite deformation analysis of the relaxed and contracted dog carotid artery. *Microvasc. Res.* 3:400–15
4. Farah JW, Craig RG, Sikarskie DL. 1973. Photoelastic and finite element stress analysis of a restored axisymmetric first molar. *J. Biomech.* 6:511–20
5. Janz RF, Grimm AF. 1972. Finite-element model for the mechanical behavior of the left ventricle. Prediction of deformation in the potassium-arrested rat heart. *Circ. Res.* 30:244–52
6. Matthews FL, West JB. 1972. Finite element displacement analysis of a lung. *J. Biomech.* 5:591–600
7. Ateshian GA. 2007. On the theory of reactive mixtures for modeling biological growth. *Biomech. Model. Mechanobiol.* 6:423–45
8. Bowen RM. 1976. Theory of mixtures. In *Continuum Physics*, ed. AE Eringen, 3:1–127. New York: Academic
9. Gu WY, Lai WM, Mow VC. 1998. A mixture theory for charged-hydrated soft tissues containing multi-electrolytes: passive transport and swelling behaviors. *J. Biomech. Eng.* 120:169–80

10. Huyghe JM, Janssen JD. 1997. Quadriphasic mechanics of swelling incompressible porous media. *Int. J. Eng. Sci.* 35:793–802
11. Lai WM, Hou JS, Mow VC. 1991. A triphasic theory for the swelling and deformation behaviors of articular cartilage. *J. Biomech. Eng.* 113:245–58
12. Mow VC, Kuei SC, Lai WM, Armstrong CG. 1980. Biphasic creep and stress relaxation of articular cartilage in compression: theory and experiments. *J. Biomech. Eng.* 102:73–84
13. Oomens CW, van Campen DH, Grootenboer HJ. 1987. A mixture approach to the mechanics of skin. *J. Biomech.* 20:877–85
14. Anderson AE, Ellis BJ, Weiss JA. 2007. Verification, validation and sensitivity studies in computational biomechanics. *Comput. Methods Biomech. Biomed. Eng.* 10:171–84
15. Henninger HB, Reese SP, Anderson AE, Weiss JA. 2010. Validation of computational models in biomechanics. *Proc. Inst. Mech. Eng. H* 224:801–12
16. Simon BR, Wu JS, Carlton MW, Kazarian LE, France EP, et al. 1985. Poroelastic dynamic structural models of rhesus spinal motion segments. *Spine* 10:494–507
17. Spilker RL, Suh JK. 1990. Formulation and evaluation of a finite element model for the biphasic model of hydrated soft tissues. *Comput. Struct.* 35:425–39
18. Suh JK, Spilker RL, Holmes MH. 1991. A penalty finite element analysis for nonlinear mechanics of biphasic hydrated soft tissue under large deformation. *Int. J. Numer. Methods Eng.* 32:1411–39
19. Donzelli PS, Spilker RL, Baehmann PL, Niu Q, Shephard MS. 1992. Automated adaptive analysis of the biphasic equations for soft tissue mechanics using a posteriori error indicators. *Int. J. Numer. Methods Eng.* 34:1015–33
20. Almeida ES, Spilker RL. 1997. 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.* 1:25–46
21. Vermilyea ME, Spilker RL. 1993. Hybrid and mixed-penalty finite elements for 3-D analysis of soft hydrated tissue. *Int. J. Numer. Methods Eng.* 36:4223–43
22. Spilker RL, Donzelli PS, Mow VC. 1992. A transversely isotropic biphasic finite element model of the meniscus. *J. Biomech.* 25:1027–45
23. Chan B, Donzelli PS, Spilker RL. 2000. A mixed-penalty biphasic finite element formulation incorporating viscous fluids and material interfaces. *Ann. Biomed. Eng.* 28:589–97
24. Wayne JS, Woo SL, Kwan MK. 1991. Application of the u-p finite element method to the study of articular cartilage. *J. Biomech. Eng.* 113:397–403
25. Ehlers W, Markert B. 2001. A linear viscoelastic biphasic model for soft tissues based on the theory of porous media. *J. Biomech. Eng.* 123:418–24
26. Laible JP, Pflaster DS, Krag MH, Simon BR, Haugh LD. 1993. A poroelastic-swelling finite element model with application to the intervertebral disc. *Spine* 18:659–70
27. Simon BR, Liabile JP, Pflaster DS, Yuan Y, Krag MH. 1996. A poroelastic finite element formulation including transport and swelling in soft tissue structures. *J. Biomech. Eng.* 118:1–9
28. Levenston ME, Frank EH, Grodzinsky AJ. 1999. Electrokinetic and poroelastic coupling during finite deformations of charged porous media. *J. Appl. Mech.* 66:323–33
29. Sun DN, Gu WY, Guo XE, Lai WM, Mow VC. 1999. A mixed finite element formulation of triphasic mechano-electrochemical theory for charged, hydrated biological soft tissues. *Int. J. Numer. Methods Eng.* 45:1375–402
30. Frijns AJ, Huyghe JM, Kaasschieter EF, Wijlaars MW. 2003. Numerical simulation of deformations and electrical potentials in a cartilage substitute. *Biorheology* 40:123–31
31. Rabbitt RD, Weiss JA, Christensen GE, Miller MI. 1995. Mapping of hyperelastic deformable templates. *Proc. SPIE Int. Soc. Opt. Eng.* 2552:252–64
32. Gardiner JC, Weiss JA. 2003. Subject-specific finite element analysis of the human medial collateral ligament during valgus knee loading. *J. Orthop. Res.* 21:1098–106
33. Weiss JA, Gardiner JC. 2001. Computational modeling of ligament mechanics. *Crit. Rev. Biomed. Eng.* 29:303–71
34. Weiss JA, Gardiner JC, Ellis BJ, Phatak NS. 2005. Three-dimensional finite element modeling of ligaments: technical aspects. *Med. Eng. Phys.* 27:845–61

35. Phatak NS, Sun Q, Kim S-E, Parker DL, Sanders KR, et al. 2007. Noninvasive determination of ligament strain with deformable image registration. *Ann. Biomed. Eng.* 35:1175–87
36. Veress AI, Klein G, Gullberg GT. 2013. A comparison of hyperelastic warping of PET images with tagged MRI for the analysis of cardiac deformation. *Int. J. Biomed. Imaging* 2013:728624
37. Weiss JA, Rabbitt RD, Bowden AE. 1998. Incorporation of medical image data in finite element models to track strain in soft tissues. *Proc. SPIE Int. Soc. Opt. Eng.* 3254:477
38. Donzelli PS, Spilker RL, Ateshian GA, Mow VC. 1999. Contact analysis of biphasic transversely isotropic cartilage layers and correlations with tissue failure. *J. Biomech.* 32:1037–47
39. Krishnan R, Park S, Eckstein F, Ateshian GA. 2003. Inhomogeneous cartilage properties enhance superficial interstitial fluid support and frictional properties, but do not provide a homogeneous state of stress. *J. Biomech. Eng.* 125:569–77
40. Ateshian GA, Ellis BJ, Weiss JA. 2007. Equivalence between short-time biphasic and incompressible elastic material response. *J. Biomech. Eng.* 129:405–12
41. Bonet J, Wood RD. 1997. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge, UK: Cambridge Univ. Press
42. Matthies H, Strang G. 1979. The solution of nonlinear finite element equations. *Int. J. Numer. Methods Eng.* 14:1613–26
43. Simo JC, Taylor RL. 1991. Quasi-incompressible finite elasticity in principal stretches: continuum basis and numerical algorithms. *Comput. Methods Appl. Mech. Eng.* 85:273–310
44. Weiss JA, Maker BN, Govindjee S. 1996. Finite element implementation of incompressible, transversely isotropic hyperelasticity. *Comput. Methods Appl. Mech. Eng.* 135:107–28
45. Puso MA, Weiss JA. 1998. Finite element implementation of anisotropic quasi-linear viscoelasticity using a discrete spectrum approximation. *J. Biomech. Eng.* 120:62–70
46. Maker BN. 1995. *Rigid bodies for metal forming analysis with NIKE3D*. Lawrence Livermore Lab. rep. UCRL-JC-119862, Univ. Calif., Berkeley
47. Simo JC, Laursen TA. 1992. Augmented lagrangian treatment of contact problems involving friction. *Comput. Struct.* 42:97–116
48. Veldhuizen TL, Jernigan ME. 1997. Will C++ be faster than Fortran? In *Proceedings of Scientific Computing in Object-Oriented Parallel Environments: 1st International Conference*, ed. Y Ishikawa, RR Oldehoeft, JWV Reynders, M Tholburn, pp. 49–56. Berlin: Springer
49. Mauck RL, Hung CT, Ateshian GA. 2003. Modeling of neutral solute transport in a dynamically loaded porous permeable gel: implications for articular cartilage biosynthesis and tissue engineering. *J. Biomech. Eng.* 125:602–14
50. Ateshian GA, Weiss JA. 2010. Anisotropic hydraulic permeability under finite deformation. *J. Biomech. Eng.* 132:111004
51. Ateshian GA, Maas SA, Weiss JA. 2010. Finite element algorithm for frictionless contact of porous permeable media under finite deformation and sliding. *J. Biomech. Eng.* 132:061006
52. Ateshian GA, Maas SA, Weiss JA. 2012. Solute transport across a contact interface in deformable porous media. *J. Biomech.* 45:1023–27
53. Ateshian GA. 2007. Anisotropy of fibrous tissues in relation to the distribution of tensed and buckled fibers. *J. Biomech. Eng.* 129:240–49
54. Ateshian GA, Rajan V, Chahine NO, Canal CE, Hung CT. 2009. Modeling the matrix of articular cartilage using a continuous fiber angular distribution predicts many observed phenomena. *J. Biomech. Eng.* 131:061003
55. Ateshian GA, Ricken T. 2010. Multigenerational interstitial growth of biological tissues. *Biomech. Model. Mechanobiol.* 9:689–702
56. Hughes TJR. 2000. *The Finite Element Method*. Mineola, NY: Dover
57. Gee MW, Dohrmann CR, Key SW, Wall WA. 2009. A uniform nodal strain tetrahedron with isochoric stabilization. *Int. J. Numer. Methods Eng.* 78:429–43
58. Puso M, Solberg J. 2006. A stabilized nodally integrated tetrahedral. *Int. J. Numer. Methods Eng.* 67:841–67
59. Maas SA, Ellis BJ, Rawlins DS, Edgar LT, Henak CR, Weiss JA. 2011. *Implementation and verification of a nodally-integrated tetrahedral element in FEBio*. SCI tech. rep. UUSCI-2011, Univ. Utah, Salt Lake City

60. Maas SA, Ellis BJ, Ateshian GA, Weiss JA. 2012. FEBio: finite elements for biomechanics. *J. Biomech. Eng.* 134:011005
61. Ateshian GA, Maas SA, Weiss JA. 2013. Multiphasic finite element framework for modeling hydrated mixtures with multiple neutral and charged solutes. *J. Biomech. Eng.* 135:111001
62. Ateshian GA, 3rd Morrison B, Holmes JW, Hung CT. 2012. Mechanics of cell growth. *Mech. Res. Commun.* 42:118–25
63. Azeloglu EU, Albro MB, Thimmappa VA, Ateshian GA, Costa KD. 2008. Heterogeneous transmural proteoglycan distribution provides a mechanism for regulating residual stresses in the aorta. *Am. J. Physiol. Heart Circ. Physiol.* 294:H1197–205
64. Roccabianca S, Ateshian GA, Humphrey JD. 2014. Biomechanical roles of medial pooling of glycosaminoglycans in thoracic aortic dissection. *Biomech. Model. Mechanobiol.* 13:13–25
65. Cortes DH, Jacobs NT, DeLucca JF, Elliott DM. 2014. Elastic, permeability and swelling properties of human intervertebral disc tissues: a benchmark for tissue engineering. *J. Biomech.* 47:2088–94
66. Weinans H, Huiskes R, Grootenboer HJ. 1992. The behavior of adaptive bone-remodeling simulation models. *J. Biomech.* 25:1425–41
67. Ateshian GA, Nims RJ, Maas SA, Weiss JA. 2014. Computational modeling of chemical reactions and interstitial growth and remodeling involving charged solutes and solid-bound molecules. *Biomech. Model. Mechanobiol.* 13:1105–20
68. Nims RJ, Cigan AD, Albro MB, Hung CT, Ateshian GA. 2014. Synthesis rates and binding kinetics of matrix products in engineered cartilage constructs using chondrocyte-seeded agarose gels. *J. Biomech.* 47:2165–72
69. Nims RJ, Cigan AD, Albro MB, Vunjak-Novakovic G, Hung CT, Ateshian GA. 2015. Matrix production in large engineered cartilage constructs is enhanced by nutrient channels and excess media supply. *Tissue Eng. C* 21:747–57
70. Albro MB, Nims RJ, Durney KM, Cigan AD, Shim JJ, et al. 2016. Heterogeneous engineered cartilage growth results from gradients of media-supplemented active TGF- $\beta$  and is ameliorated by the alternative supplementation of latent TGF- $\beta$ . *Biomaterials* 77:173–85
71. Ateshian GA. 2015. Viscoelasticity using reactive constrained solid mixtures. *J. Biomech.* 48:941–47
72. Nims RJ, Durney KM, Cigan AD, Dusseaux A, Hung CT, Ateshian GA. 2016. Continuum theory of fibrous tissue damage mechanics using bond kinetics: application to cartilage tissue engineering. *Interface Focus* 6:20150063
73. Edgar LT, Maas SA, Guilkey JE, Weiss JA. 2014. A coupled model of neovessel growth and matrix mechanics describes and predicts angiogenesis in vitro. *Biomech. Model. Mechanobiol.* 14:767–82
74. Maas SA, Erdemir A, Halloran JP, Weiss JA. 2016. A general framework for application of prestrain to computational models of biological materials. *J. Mech. Behav. Biomed. Mater.* 61:499–510
75. Hou C, Ateshian GA. 2016. A Gauss–Kronrod–Trapezoidal integration scheme for modeling biological tissues with continuous fiber distributions. *Comput. Methods Biomech. Biomed. Eng.* 19:883–93
76. Maas SA, Ellis BJ, Rawlins DS, Weiss JA. 2016. Finite element simulation of articular contact mechanics with quadratic tetrahedral elements. *J. Biomech.* 49:659–67
77. Blum MM, Ovaert TC. 2012. Experimental and numerical tribological studies of a boundary lubricant functionalized poro-viscoelastic PVA hydrogel in normal contact and sliding. *J. Mech. Behav. Biomed. Mater.* 14:248–58
78. Chen GX, Yang L, Li K, He R, Yang B, et al. 2013. A three-dimensional finite element model for biomechanical analysis of the hip. *Cell Biochem. Biophys.* 67:803–8
79. D’Lima DD, Chen PC, Kessler O, Hoenecke HR, Colwell CW Jr. 2011. Effect of meniscus replacement fixation technique on restoration of knee contact mechanics and stability. *Mol. Cell Biomech.* 8:123–34
80. Kazemi M, Li LP. 2014. A viscoelastic poromechanical model of the knee joint in large compression. *Med. Eng. Phys.* 36:998–1006
81. Langohr GD, Willing R, Medley JB, King GJ, Johnson JA. 2015. Contact analysis of the native radiocapitellar joint compared with axisymmetric and nonaxisymmetric radial head hemiarthroplasty. *J. Shoulder Elbow Surg.* 24:787–95
82. Li LP, Herzog W. 2006. Arthroscopic evaluation of cartilage degeneration using indentation testing— influence of indenter geometry. *Clin. Biomech.* 21:420–26

83. Meng Q, Jin Z, Fisher J, Wilcox R. 2013. Comparison between FEBio and ABAQUS for biphasic contact problems. *Proc. Inst. Mech. Eng. H* 227:1009–19
84. Radev BR, Kase JA, Askew MJ, Weiner SD. 2009. Potential for thermal damage to articular cartilage by PMMA reconstruction of a bone cavity following tumor excision: a finite element study. *J. Biomech.* 42:1120–26
85. Warner MD, Taylor WR, Clift SE. 2001. Finite element biphasic indentation of cartilage: a comparison of experimental indenter and physiological contact geometries. *Proc. Inst. Mech. Eng. H* 215:487–96
86. Wu JZ, Herzog W, Epstein M. 1998. Evaluation of the finite element software ABAQUS for biomechanical modelling of biphasic tissues. *J. Biomech.* 31:165–69
87. Federico S, La Rosa G, Herzog W, Wu JZ. 2004. Effect of fluid boundary conditions on joint contact mechanics and applications to the modeling of osteoarthritic joints. *J. Biomech. Eng.* 126:220–25
88. Galbusera F, Bashkuev M, Wilke HJ, Shirazi-Adl A, Schmidt H. 2014. Comparison of various contact algorithms for poroelastic tissues. *Comput. Methods Biomech. Biomed. Eng.* 17:1323–34
89. Pierrat B, Murphy GJ, Macmanus DB, Gilchrist MD. 2015. Finite element implementation of a new model of slight compressibility for transversely isotropic materials. *Comput. Methods Biomech. Biomed. Eng.* 19:1–14
90. Erdemir A. 2013. OpenKnee: a pathway to community driven modeling and simulation in joint biomechanics. *J. Med. Devices* 7:0409101
91. Erdemir A. 2015. OpenKnee: open source modeling and simulation in knee biomechanics. *J. Knee Surg.* 29:107–16