# A Systematic Comparison Between FEBio and PolyFEM for Biomechanical Systems

Liam Martin[a], Pranav Jain[b], Zachary Ferguson[b], Torkan Gholamalizadeh[c], Faezeh Moshfeghifar[d], Kenny Erleben[d], Daniele Panozzo[b], Steven Abramowitch[a] and Teseo Schneider[e,*]

[a]University of Pittsburgh Swanson School of Engineering, USA
[b]New York University, USA
[c]3Shape A/S, Denmark
[d]University of Copenhagen, Denmark
[e]University of Victoria, Canada

ARTICLE INFO

*Keywords*:
Finite element analysis
Large deformation
Contact
Benchmark
Finite element verification

ABSTRACT

**Background and Objectives.** Finite element simulations are widely employed as a non-invasive and cost-effective approach for predicting outcomes in biomechanical simulations. However, traditional finite element software, primarily designed for engineering materials, often encountered limitations in contact detection and enforcement, leading to simulation failure when dealing with complex biomechanical configurations. Currently, a lot of model tuning is required to get physically accurate finite element simulations without failures. This adds significant human interaction to each iteration of a biomechanical model. This study addressed these issues by introducing PolyFEM, a novel finite element solver that guarantees inversion- and intersection-free solutions with completely automatic collision detection. The objective of this research is to validate PolyFEM's capabilities by comparing its results with those obtained from a well-established finite element solver, FEBio.

**Methods.** To achieve this goal, five comparison scenarios were formulated to assess and validate PolyFEM's performance. The simulations were reproduced using both PolyFEM and FEBio, and the final results were compared. The five comparison scenarios included: (1) reproducing simulations from the FEBio test suite, consisting of static, dynamic, and contact-driven simulations; (2) replicating simulations from the verification paper published alongside the original release of FEBio; (3) a biomechanically based contact problem; (4) creating a custom simulation involving high-energy collisions between soft materials to highlight the difference in collision methods between the two solvers; and (5) performing biomechanical simulations of biting and quasi-stance.

**Results.** We found that PolyFEM was capable of replicating all simulations previously conducted in FEBio. Particularly noteworthy is PolyFEM's superiority in high-energy contact simulations, where FEBio fell short, unable to complete over half of the simulations in Scenario 4. Although some of the simulations required significantly more simulation time in PolyFEM compared to FEBio, it is important to highlight that PolyFEM achieved these results without the need for any additional model tuning or contact declaration.

**Discussions.** Despite being in the early stages of development, PolyFEM currently provided verified solutions for hyperelastic materials that were consistent with FEBio, both in previously published workflows and novel finite element scenarios. PolyFEM exhibited the ability to tackle challenging biomechanical problems where other solvers fell short, thus offering the potential to enhance the accuracy and realism of future finite element analyses.

## 1. Introduction

Simulations of biomechanical systems are often used as a controlled and cost-effective way to make predictions of normal and/or pathological processes, to gain insights into these complex systems through parametric analyses, to design devices, as an indirect and non-invasive way to perform measurements, and as a way to communicate and educate [68, 24, 33, 2, 54, 62, 10]. Traditionally, computational biomechanics, and bioengineering in general, have benefited significantly from adapting theories and approaches developed to solve traditional engineering problems with traditional materials. For example, rubber elasticity provided an excellent general framework for understanding the fundamentals of tissue mechanics. However, many of these tools were never designed specifically to solve problems in biomechanics so they often fail to sufficiently describe specific aspects of biological mechanical behavior that are often required to answer specific biological questions (e.g., rubber elasticity cannot describe tissue growth and remodeling) [32].

Energy transfer via contact and friction is particularly challenging for simulations and proves to be especially problematic in the context of biological tissues. Compared to standard engineering materials, biological tissues can undergo very large non-linear deformations, even in response to relatively small forces, and are often in contact with other tissues that are mutually deformable. Small errors in the calculation of forces can result in very large deformations

*Corresponding author
ORCID(s):

that do not accurately simulate the system. Thus, it is not only important to accurately describe material behaviors in these scenarios, but it is also critical to accurately describe mechanical interactions between materials that share contact surfaces.

For most scenarios, there are a few common configurations that are particularly challenging:

1. thin, soft layers compressed between large and stiff objects (for example, cartilage and menisci),
2. high-energy collisions,
3. large deformations of soft tissues,
4. complex contact between multiple objects in close proximity.

In all these cases, there are often failures due to either individual elements degenerating into zero or negative volume (often referred to as negative Jacobian elements) or an inability to correctly resolve collisions leading to either invalid simulation states or non-physical impulse forces to compensate for the incorrect collision response. These problems are tackled in existing simulators by providing parameters that allow controlling both contact and elastic forces to prevent these configurations. However, finding a valid set of parameters for scenes with complex geometries and scenarios can be extremely challenging and time-consuming. Furthermore, there is no guarantee that a set of parameters even exists. This can lead to an infinite loop of adjusting parameters that may ultimately never produce a viable result. Once this happens, the user either has to make compromises (e.g., changes to the geometries, altering the boundary conditions, or otherwise simplifying the simulation) in order for the simulation to complete.

A new family of robust FE solvers based on the Incremental Potential Contact (IPC) formulation [40] has been recently introduced for structural mechanics problems: the key difference in these approaches is that their formulation is, by construction, addressing the two issues above. No element can invert, and no collision can be missed. This is achieved with an entirely different (and not equivalent) formulation, which trades off computational efficiency for increased robustness and reduction of parameter tuning. In this work, we benchmark the implementation of this approach in the PolyFEM [60] open-source software to evaluate its utility for biomechanical simulations, comparing it against the established FEBio software [43]. Each of the tests in the benchmark compares different simulation's outcomes, including stresses, strains, and displacements. As there is no clear definition of equivalence between different results, for this study, we deemed the solvers to be equivalent if the difference in the outcome's measure is less than 5%. However, many of the simulations, especially those without contact, produce identical results because the solvers are based on the same material models.

We observed that the results obtained by PolyFEM are very similar to FEBio while requiring much less parameter tuning; in some complex cases, we found that PolyFEM was able to simulate systems that proved to be challenging for FEBio. On the other hand, PolyFEM is still in the early stages of development and thus does not yet support a wide selection of features that are necessary for many biomechanical simulations, including reduced models of rods and shells, advanced material models, and certain constraints. As noted, it is important to recognize that the ability to handle more complex simulations also comes at a higher computation price; based on our experience, we believe this is a fair tradeoff, as computational resources are affordable compared to the human effort required for parameter tuning.

## 2. Related Works

### 2.1. Biomechanics Simulations

We note that the list of FE studies and software included in this section is by no means exhaustive. Providing such an exhaustive review is beyond the scope of this work; however, we believe that it is important the contextualize our work by providing a representative selection of other software that is often used in biomechanics research.

A common application for the use of specialized simulation is in the area of musculoskeletal modeling. Software for these simulations is based on using rigid multi-body systems for bones and Hill-based (spring-like) models for muscles [61, 13]. While very important and successful for many questions related to joint kinematics and dynamics, muscle force estimation, and muscle activation patterns, such simulators ignore inter-contact between muscles and model muscle-bone interaction directly via points. The type of problems addressed often implies inverse dynamics and contact with the environment are prescribed as boundary constraints. Hence, they often do not include the elasticity of tissue and use idealized assumptions on joints and contact, sometimes driven by real-life force measurements. It is not uncommon to use simulation outputs from such simulators to estimate forces that can drive motions in finite-element simulations.

For fast solvers for real-time medical simulations, there exist frameworks such as SOFA [21] which are well designed to provide solutions for pre-guided image surgery, control of soft medical robots, surgical training, and more. SOFA focuses on performance to deliver fast real-time interaction with clinical operators [74]. By using the finite element method with a focus on linear elements and co-rotational linear elastic materials mixed with optimizations of matrix computations that exploit zero-fill patterns, this software can achieve significant performance gains at the cost of accuracy. In terms of contact, the SOFA does support general collision detection and implements constraint-based contact forces using expressed LCP models based on the classic Coulomb friction models for planar dry friction. Nevertheless, these compromises in accuracy in favor of performance are often justified for some problems in biomechanics. SOFA can also be extended. For example, the inverse finite element method is being used in SOFA

to support control of soft medical robots [45]. In addition, FEniCSx and SOFA have also been combined [44] providing SOFA with advanced FE features and support for users to implement their constitutive model of choice through coding both for direct forward and inverse simulations.

On the other hand, many problems in biomechanics often necessitate accuracy on a level that cannot be provided by fast real-time medical simulators. These simulations are usually performed using commercial finite element (FE) software packages (e.g., Ansys$^{TM}$ [15, 52, 56], ABAQUS$^{TM}$ [36, 29, 78, 42], COMSOL [26, 27], and NIKE3D$^{TM}$) or open-source solvers like FEBio (University of Utah, and Columbia University) [43, 23, 58, 39] or aforementioned FEniCSx [41]. These solvers have largely evolved from traditional structurally focused engineering solvers, and while they do provide state-of-the-art material models for biomechanics and are often robust to handle many biomechanical scenarios, they were not specifically designed to capture some of the complex mechanical interactions that are common in biomechanics (e.g., large deformation, sudden contact, and friction forces). As such, the contact models are generally most suitable for structural mechanics applications. While these can be effective for specific biomechanical applications (e.g., orthopedics), they often require a large degree of parameter tuning and often explicit specification of the contact surfaces. This can present significant challenges for simulating soft tissue-to-soft tissue interactions with nonuniform geometries that undergo major changes in shape, size, and areas of contact. Even for well-posed problems, incorrect parameter choices can often lead to simulation failure or inaccurate results. Other solvers, such as the SIMon Finite Element Head Model (developed in part by the National Highway Traffic Safety Administration), are designed to simulate specific scenarios; namely head trauma in motor vehicle accidents [65, 66]. Other studies either used less popular software [64, 57, 16] or did not explicitly state which FE solver they used [20, 75].

## 2.2. Existing Benchmarks of Finite Element Solvers

We are not aware of a comprehensive set of benchmarks that can evaluate an FE solver's ability to compute complex biomechanical problems. Therefore, the responsibility falls on the software developers and model creators to ensure the accuracy of their work. FE benchmarks can be broadly divided into two categories, verification and validation. The former focuses on confirming that the solver produces accurate mathematical solutions, while the latter involves ensuring that the computational model accurately simulates real-world physical interactions [3, 28].

In the past, verification has primarily been the responsibility of the solver's creators, who have released verification problems along with their FE solver. These problems serve to demonstrate that the underlying mathematical implementation is sound by comparing the solver's solutions to known analytical solutions and/or previously verified FE solvers [1, 4, 43]. Although some groups have attempted to compile a comprehensive list of verification problems that should accompany any FE solver, these lists have yet to gain significant adaptation [51, 17, 47, 18]. The most common verification problems are simple simulations with well-known analytical solutions and will be presented in more detail later in the paper (i.e., a cantilever beam, hyperelastic sheet with hole, single element tension/compression, etc.). This study's major focus was on this topic, to ensure that the underlying mathematical implementation of material models and boundary conditions within PolyFEM are correct.

Validation, on the other hand, is usually produced to accompany the release of a FE model. In these benchmarks, the model's creator should attempt to prove that their model is capable of modeling real-world physical interactions. In biomechanics, this typically involves one or more of the following: comparing the measures generated by an FE model to experimental biomechanical data, such as stress, strain, and displacement [15, 65, 66, 76, 52, 22, 16, 64, 78, 56], cadaveric and/or human system measures [65, 76, 22, 36, 52, 23, 58, 39], or even other FE solvers [36, 20, 56]. In cases where simple outcome measures are nearly impossible to measure (i.e., *in vivo* tissue response), comparing the motion of organs/tissues on dynamic MRI to that calculated from the model has been used as proxy [57, 42]. As previously stated the focus of this study was verification of PolyFEM's mathematical implementation however, some of the examples are based on analytical solutions or physiologic data (sections 4.1, 4.2, 4.5, and 4.6) and thus some validation may be possible. Any future work creating a model in PolyFEM, as is true with any finite element solver, would need to be further validated for their specific model through one of the aforementioned methods

Beyond comparing the accuracy of FE solutions, benchmarks offer the ability to compare the efficiency of FE software while simulating the mechanical problem. Assuming that the solvers produce identical stress and strain states, the easiest of these comparisons to make is the CPU time that it takes for the solvers to converge to the same solution. This notably does not include the time that it takes for the user to set up the simulation, or "human time," which in most cases is the most time-consuming portion of FE model development. Human time also extends to the iterative process where the user has to adjust model parameters (meshes, contact penalties, etc.) in order for the model to converge to a solution. There are not many studies that aim to determine which FE software is the fastest. Those that do compare solvers are comparing specific components of the software like the contact algorithm or solving method [46, 38]. One of the potential reasons for this lack of study is that the majority of researchers in this field will choose the FE software that they are most comfortable with, even if there are potentially significant time delays in doing so.

## 2.3. Common Contact Models in Biomechanics

Biomechanical simulations often require the accurate modeling of physical interactions (i.e., contact) between different tissues, such as those that occur in joints, organ

systems, foot/ground interactions, and others. Detecting and implementing methods to resolve the transfer of energy during these interactions are some of the most challenging areas in biomechanical simulation. In general, three classes of contact have been used to detect and implement contact; node-on-node, node-on-segment, and segment-on-segment [73]. Node-on-node contact can only be used in linear cases with symmetrical meshes and thus will not be discussed further. Node-on-segment contact was first developed to address a common problem in all contact methods, i.e. penetration between the two objects that have entered contact with each other. This is handled by first checking for, and if needed, addressing, intersecting faces [73, 55, 67, 31, 63, 71]. Addressing these intersections depends on the solver that is used and will be discussed later. A single pass node-on-segment approach only requires that the nodes from one object (object A) do not intersect with the faces of another object (object B), also known as ”primary and secondary” surfaces [55]. Two-pass approaches do the same thing as single-pass but also ensure that the nodes from B also do not intersect the faces from A [55]. However, these methods are prone to four major drawbacks, which are discussed in much further detail in Puso et al. and Erleben [55, 19]:

1. Locking or over-constraint of some nodes
2. Non-smooth contact that leads to jumps in contact forces when nodes from an object slide between the faces of the other
3. The discrete constraints cause jumps when nodes from one object slide off the boundary of the other
4. Inequality equations determine active and inactive constraints

These four drawbacks were the significant driving force behind the development of surface-on-surface algorithms, which can address the top three drawbacks [55]. By using smoothed surface approximations to calculate contact, these algorithms avoid the possibility of nodes getting "locked" in place or experiencing significant jumps due to sliding between surfaces or off of the boundary of the surface. Most software allows the user to select which of these contact detection formulations they want to use. Then the method for enforcing the detected contact is software-dependent. Several algorithms have been developed for enforcing contact, and two widely used methods are the penalty and Lagrange multiplier methods [8, 50]. In general, both methods apply constraints that limit the possibility of infeasible solutions forming, i.e., intersection detected between two objects. The augmented Lagrangian method uses the principals of both aforementioned methods but also includes additional augmentation steps to improve the estimates of the Lagrangian multipliers and is implemented in popular software packages such as FEBio, ANSYS, and ABAQUS [70]. The augmented Lagrangian method starts with a penalty step and then enters an augmentation cycle where the Lagrangian multipliers are iteratively updated to improve the estimates of the multipliers. These methods are easier to implement than others we will discuss because they only add a multiplier to the objective function [7]. However, their simplicity can introduce significant bias to the simulation since the choice of penalty is often arbitrary and can significantly impact the outcome [6]. Although these methods work well in simple contact cases, they often struggle when computing high-energy contact between soft deformable bodies, such as human organs.

Other, less popular biomechanical FE software derive their regularized contact model from Nitche’s method. One such software is CutFEM [9, 12]. CutFEM has been designed to make the problem’s discretization as independent as possible from the geometric description and to minimize the complexity of mesh generation while maintaining the accuracy of the FE method [9]. Contact interfaces between two meshes are represented by a level set function that is placed on a background grid of the simulation, which allows for low-quality and/or complex geometries to be modeled without the need for computationally expensive remeshing. By using this discretization method, it becomes much easier to implement Nitsche’s method for contact [12]. Nitsche’s method and its derived regularized contact models apply a penalty term to the weak form of the governing equations and can be viewed as a generalization of the classic penalty model. However, unlike the classical penalty model, Nitsche’s method is symmetric and consistent across boundaries, which works well with CutFEM’s implementation of geometric boundaries. Symmetry across boundaries ensures that these methods do not suffer from any of the aforementioned locking or jumping effects. Unfortunately, in nonlinear cases, Nitsche’s method becomes more complex than penalty or Lagrange multiplier methods and thus more challenging to compute. This becomes problematic in biomechanical simulations as the majority of them include some sort of nonlinear contact [71]. Additionally, Nitche’s method uses a penalty parameter that must be arbitrarily assigned and has a significant effect on the simulation outcome.

The final type of contact models to be discussed are those based on barrier stiffness methods. These methods are utilized in PolyFEM, which employs the IPC contact library [40]. A barrier stiffness model operates by introducing a stiffness term that prohibits two contacting bodies from intersecting. At the time of their publication Li et al. stated that IPC is the first implementation of a contact model that can ensure convergence of solutions free of intersections and inversions (which, based on another literature search, appears to remain true) [40]. We are not claiming that this is the first implementations of barrier stiffness methods for biomechanics simulations. In fact these methods have become more and more popular over the last 20 years [37, 15, 34, 59]. If a reader is interested in how IPC fits in the landscape the history of other barrier stiffness methods we point the reader to Li et al. and Laursen [37, 40] This makes barrier stiffness methods particularly suitable for problems with significant nonlinear deformations, such as those encountered in biomechanical simulations. However,

it should be noted that the suitability of this software for biomechanical simulations has not yet been verified, which is something we will aim to assess in subsequent sections.

## 3. Mathematical Background

We briefly overview the major solver differences between FEBio and PolyFEM, focusing on their relevance in biomechanics. We exclude from our discussion friction forces; we refer an interested reader to Maas et al. [43] and Li et al. [40] for more details. The major difference between the two solvers is that PolyFEM expresses all parts (elasticity, inertia, contact, etc.) as potentials, while FEBio uses only the elastic energy. While both formulations are mathematically equivalent, the PolyFEM formulation allows using a standard unconstrained optimization method coupled with a line search to ensure that the solution remains in the feasible region, thereby having the capability of handling challenging cases such as small elements being compressed or high velocities leading to large deformations. Granted this will lead to a harder minimization problem that might require more iterations for the quadratic approximations of the Newton solver; however, the method is inherently robust as it is guaranteed to produce a physically valid configuration for any provided displacement or velocity (i.e., the solver remains in the feasible region).

*Elastic Potential.* Both FEBio and PolyFEM use the same elastic potential $E_e$ derived from the elastic energy. However, FEBio supports significantly more material models; for instance, transversely isotropic (Transversely Isotropic Hyperelastic) and orthotropic (Fung Orthotropic, Holzapfel-Gasser-Ogen) materials are not yet implemented in PolyFEM. The major advantage of the potential formulation in PolyFEM is that in the line search, it explicitly checks for inverted elements and shortens the Newton increment to ensure that the new solution is valid. This occurs since the quadratic approximation of $E_e$ used by the solver does not diverge when elements have zero volume, even if $E_e$ does. We show an example of such a problem in Figure 2. While this may seem like a minor change, it is possible to implement it only because of the difference in how the solver is set up.

*Inertia Potential.* FEBio implements the standard time integration scheme[1] while PolyFEM uses the incremental potential formulation [35]. Both formulations are equivalent and support several standard time integrators (e.g., Newmark or backward differentiation formula). In PolyFEM, the inertia potential is simply summed to the elastic potential.

*Contact Potential.* While the previous potentials (elastic and inertia) are identical, PolyFEM and FEBio handle contact differently. From a high-level, point of contact requires a set of nodal positions $x^t$ and nodal velocities $v^t$, a choice of spatial and temporal discretization, and a measure of overlap between primitives $g(x)$, and obtains the updated nodal positions by solving a constrained minimization of a potential $E$ [35] (inertia and elasticity in PolyFEM and elasticity in FEBio):

$$x^{t+1} = \operatorname*{argmin}_x E(x, x^t, v^t), \quad \text{s.t. } g(x) \geq 0. \tag{1}$$

The choice of $g$ varies, but it is usually a function that is zero when elements do not overlap and negative otherwise. There are many ways of defining; for instance, FEBio uses the signed distance along the normal direction between the closest points [77]. This problem is typically solved using off-the-shelf or custom numerical solvers; FEBio uses a Newton-Raphson method [43]. As for the elastic potential $E_e$, the solution of (1) with linearized constraints does not directly imply that $g(x) \geq 0$, and even solving a sequence of problems with linearized constraints at each step might not necessarily find a valid configuration satisfying the non-linear constraints, thus potentially not resolving collisions. We show an example of such failure in Figure 10. Another source of failure is that constrained solvers usually only satisfy the constraints up to numerical precision. This might lead to missed/problematic collisions when large or small numbers are present (e.g., in the presence of high velocities or small elements).

*Incremental potential contact.* The IPC formulation tackles these failure points by avoiding the use of constrained solvers and making the linearization of constraints and energy safe by using a custom line search procedure, as for $E_e$. The constrained optimization problem (1) is converted into the unconstrained optimization of:

$$B_t(x) = E(x, x^t, v^t) + \kappa \sum_{k \in C} b(d_k(x)),$$

where $\kappa > 0$ is an adaptive parameter controlling the barrier stiffness, $d_k$ measures the distance between two primitives in the set of all possible primitive pairs $C$, and $b$ is a logarithmic barrier function. This non-linear energy is minimized with a Newton descent algorithm with a custom line search that explicitly prevents crossing configurations where $B_t(x)$ is infinite; that is, when two primitives are at zero distance (i.e., there is an overlap between two primitives). These two conditions are tested with algorithms that are exact under floating point rounding [69].

## 4. Methods and Results

We directly compared each of the two solvers in five head-to-head comparison tests. We selected the first two benchmarks, *FEBio-Test* (Section 4.1) and *FEBio-Verification* (Section 4.2), to validate the capability of PolyFEM to conduct simulations similar to those previously published by FEBio [5, 43]. Using the information from these two sections and the previously published FEBio verification paper, we conducted general comparisons with other finite element solvers, such as ABAQUS, which had previously been compared to FEBio [43]. We designed the third benchmark to emphasize PolyFEM's potential as a finite element

[1] https://help.febio.org/docs/FEBioTheory-4-0/TM40-Chapter-6.html

software in complex biomechanics simulations (Section 4.3) by including complex contact between elastic bodies in a scenario based on biomechanics. Additionally, the fourth comparison, *Planet-Fall* (Section 4.4), introduced a high degree of non-linearity into the simulation while exploring the limits of material models and the solvers themselves. Finally, the fifth benchmark, *Hip-and-Jaw* (Section 4.5), serves as a real-world example, using patient data, of using PolyFEM as a biomechanics solver [49, 25]. We conducted all simulations using the same version of each respective solver regardless of the test. For FEBio, we used the pre-compiled version of FEBio studio (version 1.8) (FEBio Studio Download, FEBio solver version (3.7.0)), and for PolyFEM, we used version 1.0 from their GitHub repository [60]. For each of the verification problems below, the time steps were set to be the same between the two solvers. However, some simulations required dynamic time-stepping to be enabled in FEBio to complete the simulations which led to slightly different time-steps between the two solvers for some simulations.

### 4.1. FEBio Test Suite Examples

The test suite is a set of examples that outline the features of FEBio, including static, dynamic, and contact simulations. We selected a group of 18 problems from the test suite, while also adding a new one, and simulated them all using both solvers. Of the 18 selected simulations, 9 are static and involve contact, while the other 9 are dynamic (Figure 1 shows three frames for 7 examples). We ran our simulations on AMD Ryzen Threadripper PRO 3995WX, 64 Core (2.7GHz) sWRX8 Processor, 2TB 3200 MHz DDR4 memory, Ubuntu 22.04.1 LTS using 16 threads. Depending on the simulation, outputs for comparisons included displacement or stress, and simulation time in terms of performance. Both FEBio and PolyFEM produce similar results; but PolyFEM, in general, takes more time to simulate. Table 1 lists the time taken by PolyFEM and FEBio on the selected simulations. We accounted for these timings being computer-specific in this manuscript by employing the same computer for both solvers within each of the verification tests. For all of the simulations in this benchmark, the average point-to-point distance between the solvers was much less than the 5% we allowed. For example, in simulation co20 (Figure 1 second row, Table 1 third row) the average distance between all points at all time steps in the simulation was 0.0030 m, or 0.15% of the shortest axis of the block (maximum 0.0484 m, 2.42%). We note that all of the tested FEBio TestSuite problems that involved contact (all of the "co" problems, and dy03, dy04, and dy09) used settings other than the default for the contact interface while the same simulations in PolyFEM were simply converted and worked with the default contact settings. The two cases in which PolyFEM was faster compared to FEBio (co21 and dy09) are notable because they are the only simulations within the TestSuite where PolyFEM outperformed FEBio. An explanation for

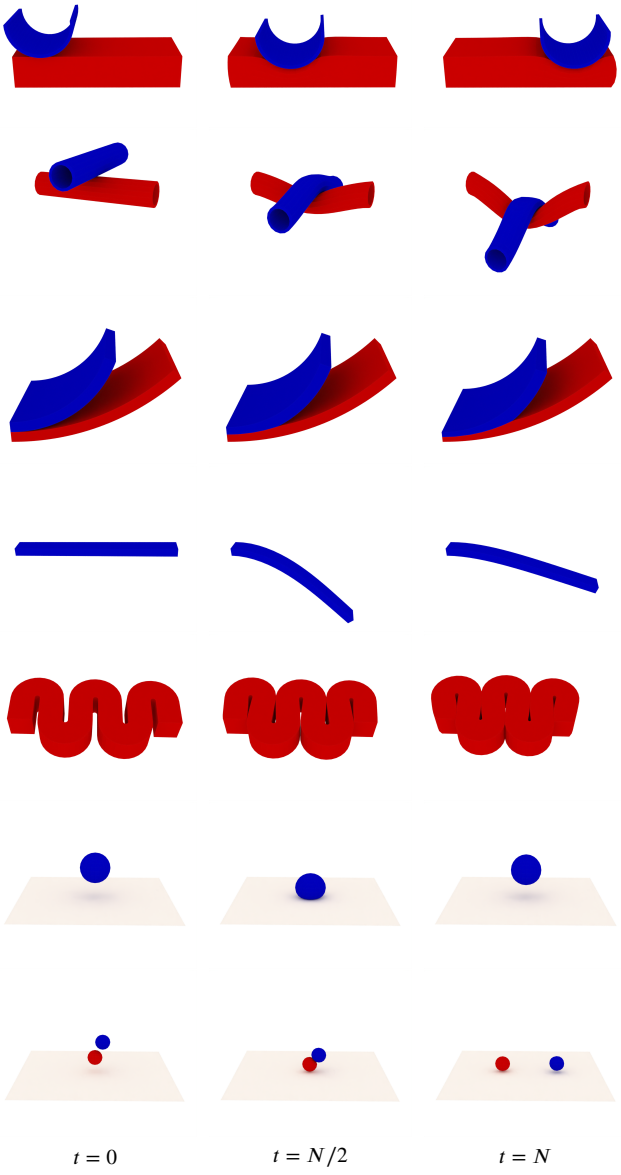



**Figure 1:** Results of 7 simulations (co07, co20, co34, dy02, co41, dy03, and dy04, from top to bottom) from the FEBio-Test dataset. Each row corresponds to one example simulated from $t = 0$ to $t = N$. Images in the figure were generated using PolyFEM visualized in Paraview.

this outcome could be attributed to the fact that these particular simulations consisted of complex sliding (co21) or high-energy collisions (dy09). Such complex contact scenarios may be handled better using PolyFEM's contact algorithm. However, significant further investigation and analysis must be conducted to confirm these hypotheses.

To highlight the differences in the limits of the solver's performance when modeling large deformations of very soft elastic materials, we created a new simulation using geometry from the existing benchmark, dyn02, a 20 m × 2 m × 1 m rectangular block in a beam bending scenario. The goal of this simulation was to introduce a negative jacobian due to

**Table 1**
Timings for the simulations from FEBio-Test dataset.

| Simulation | Category | PolyFEM (s) | FEBio (s) |
|---|---|---|---|
| co07 | Static | 13.237 | 12.1 |
| co16 | Static | 4.122 | 0.047 |
| co20 | Static | 66.811 | 52.169 |
| co21 | Static | 15.550 | 52.764 |
| co32 | Static | 0.280 | 0.136 |
| co34 | Static | 30.114 | 0.412 |
| co35 | Static | 91.887 | 0.095 |
| co41 | Static | 57.223 | 5.794 |
| co44 | Static | 4.371 | 0.058 |
| dy01 | Dynamic | 0.330 | 0.015 |
| dy02 | Dynamic | 10.663 | 0.259 |
| dy03 | Dynamic | 29.567 | 23.596 |
| dy04 | Dynamic | 38.474 | 5.667 |
| dy07 | Dynamic | 2.918 | 0.126 |
| dy09 | Dynamic | 181.853 | 332.919 |
| dy10 | Dynamic | 0.176 | 0.023 |
| dy11 | Dynamic | 0.477 | 0.024 |
| dy12 | Dynamic | 1.407 | 0.031 |

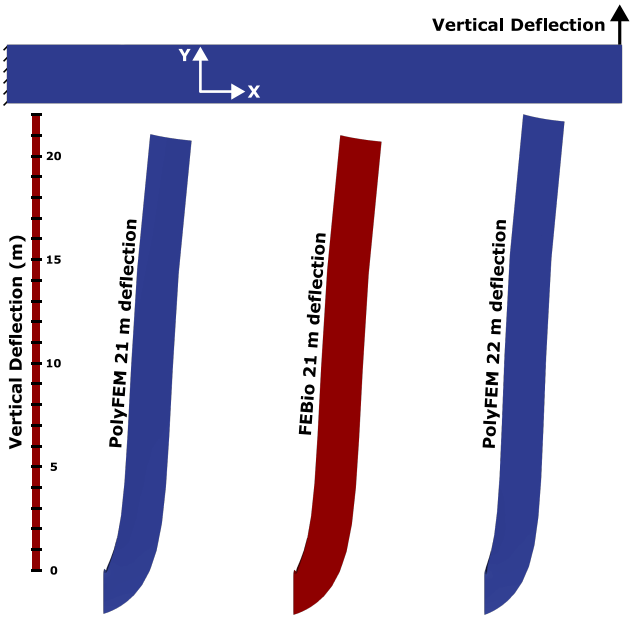



**Figure 2:** A 20 m × 2 m × 1 m rectangular block deflected in the $y$ direction from the right-hand side. The first figure shows the setup in its undeformed configuration, and the second and third image shows the FEBio and PolyFEM results, respectively, for a deflection of 21 m. The final picture shows the results for a deflection 22 m using PolyFEM, as FEBio cannot find a solution.

elastic stretching in a simple mechanics problem. Errors like this one are a common failure observed in computational biomechanics and do not have a guaranteed solution. The simulation was first changed to be a static simulation and he beam was oriented parallel to the world's x-axis. We modeled the material as Neohookean with $E = 100\,\text{Pa}$, $\nu = 0.0$, and $\rho = 1$) and meshed it within FEBio using tetrahedral elements ($Nx = 20$, $Ny = 10$, $Nz = 4$, $X - bias = 1.4$, $Y - bias = 1$, $Z - bias = 1$). We choose tetrahedral elements because of their common use in biomechanics simulations. We fixed the left side along all three axes (negative $x$ direction) and limited the front side from deflecting into/out of the page (positive $z$ direction). On the right side of the block (positive $x$ direction), we prescribe a vertical displacement (in the positive $y$ direction) until FEBio failed using the default simulation settings, which occurred at 21 m of vertical deflection.

We note that changing the mesh elements to hexahedral elements or changing the mesh bias, allows for more beam deflection. We then replicated the same simulation in PolyFEM but prescribed a vertical deflection of 22 m (one meter more than what FEBio was able to handle). It is important to note that both solvers produced the same solution for 21 m of vertical deflection, which is shown in Figure 2. Although this is an extreme example designed to produce a failed solution in FEBio, it highlights PolyFEM's ability to handle large hyperelastic deformations without inverting elements.

## 4.2. FEBio Verification Examples

Our second set of comparisons was based on the verification paper released by FEBio, which compares the results of FEBio to analytical results, as well as results generated with ABAQUS$^{TM}$ and NIKE3D$^{TM}$ [43]. The paper outlines ten simulations and verifies the results from FEBio with respect to the other two solvers. Thus, the following section not only serves to verify the results from PolyFEM with FEBio but also the other two solvers by extension. All simulations were conducted using PolyFEM and FEBio using the same computer by the same author (2017 iMac Pro, 10 Core (3GHz), all of which were allocated for this problem, Intel Xeon W, 128 Gb 2666 MHz DDR4 memory, macOS Ventura 13.0.1).

The initial example uses a single hexahedral element mesh and compresses it to 0.5 times, and stretches it to 1.5 times its original length in one axis. As the dimensions of the geometry were not specified by [43], we choose a 1 mm × 1 mm × 1 mm cube. We evaluated the resulting stress inside the element using two different material models, namely the Mooney-Rivlin ($C_1 = 6.8\,\text{MPa}$, $C_2 = 0$, $K = 100\,\text{GPa}$) and the Ogden hyperelastic ($N = 1$, $c_1 = 0.0329\,\text{MPa}$, $m_1 = 6.82$). Each of the simulations, in both solvers, finished simulating in under 0.5 s. We ploted the von Mises stress (for both material models) to compare the results between FEBio and PolyFEM (Figure 3); both solvers produce the same values in tension and compression.

The next example modeled a hyperelastic sheet placed under extreme deformations. The sheet's undeformed dimensions are 165 mm × 165 mm × 2 mm with a circular hole ($r = 6.35\,\text{mm}$) cut in the center of the large face (Figure 4 top). We meshed it with 128 hexahedral elements and modeled the material as Mooney-Rivlin ($C_1 = 0.1863\,\text{MPa}$, $C_2 = 0.009\,79\,\text{MPa}$, $K = 100\,\text{MPa}$). The top and bottom faces are not allowed to move in the $y$ direction, and the front and back faces are not allowed to displace in the $z$ direction.

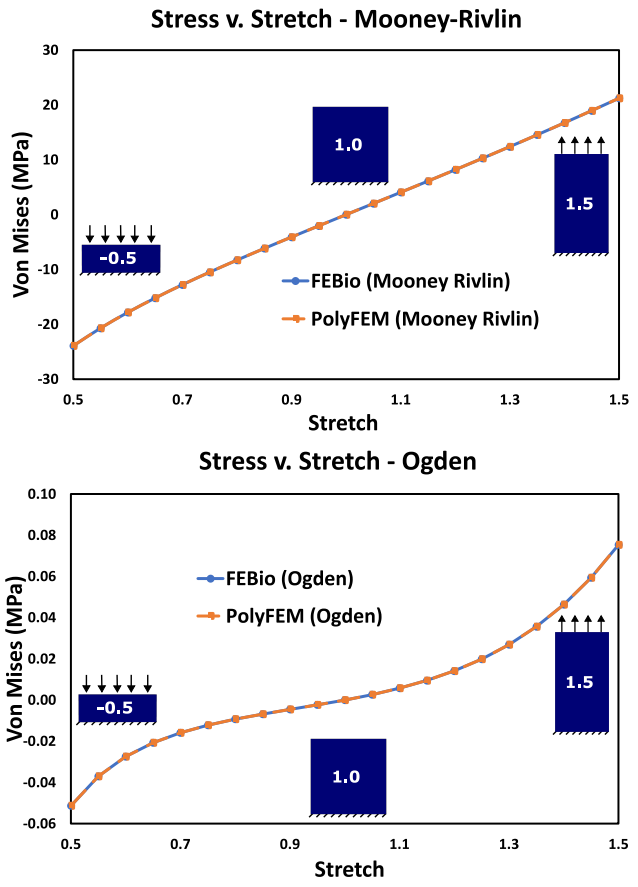



**Figure 3:** Comparison of the von Mises stress versus stretch between PolyFEM and FEBio for a single hexahedral element modeled with two material models.

We compared the reaction forces on the left face between the two software packages. We note that reaction forces are calculated slightly differently between the two software packages. In FEBio, the reaction forces are calculated at each node, while PolyFEM computes them on the surface (i.e., traction force). While FEBio Studio can integrate forces over a surface we chose to convert the traction force calculated in PolyFEM to FEBio. To convert between FEBio and PolyFEM, we needed to multiply PolyFEM's force by the cross-sectional area of the element. We found that the two reaction forces were slightly different in value (less than 3%). This difference is likely due to differing implementations of the Mooney-Rivlin material model. However, it should be noted that the reaction force reported by PolyFEM matched the force calculated by ABAQUS. Similar to most of the other simulations, FEBio was significantly faster than PolyFEM (2 and 56 seconds, respectively).

In the next simulation, we applied a load that induced approximately 8 m of lateral deflection to a 0.15 m×0.10 m× 10 m long rectangular cantilever beam. We meshed it with 400 hexahedral elements along its length and one through, each its width and depth. We fixed one face at the end of the length and applied a load 269.35 N to the tip at the other end, which was perpendicular to the depth of the beam.

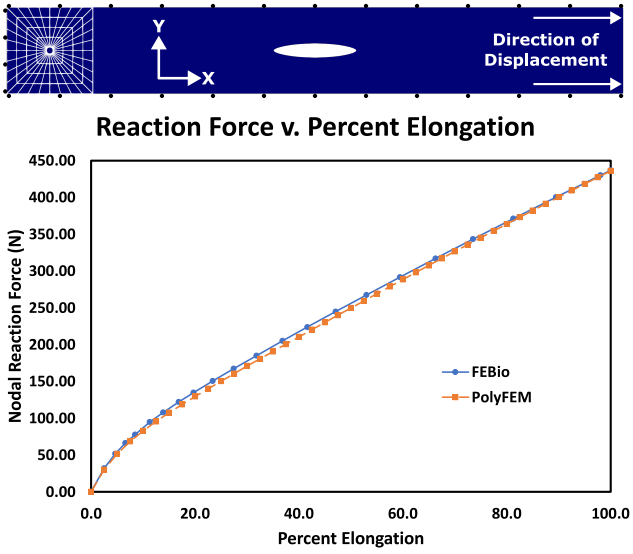
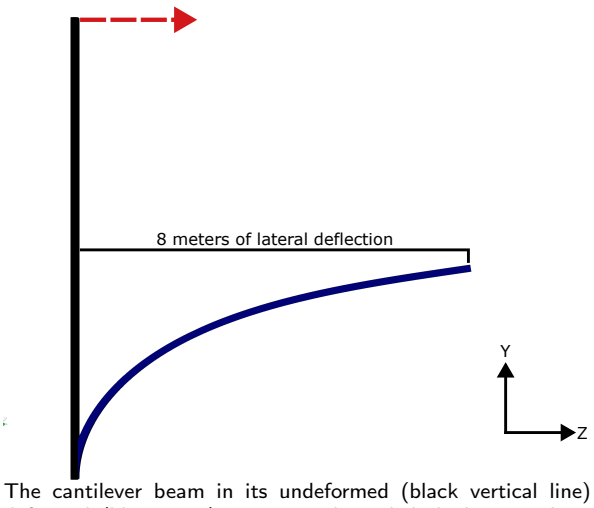


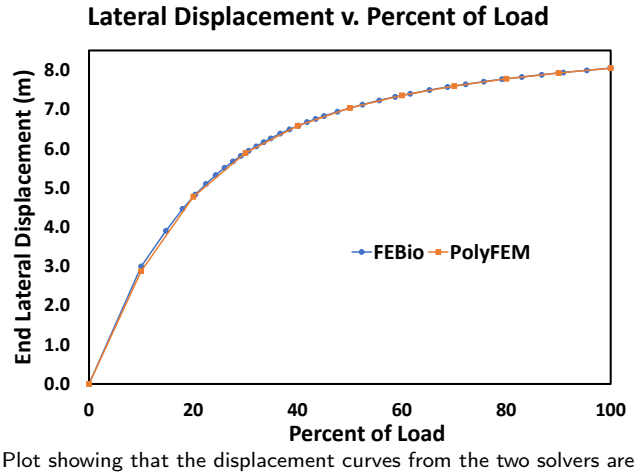

**Figure 4:** Hyperelastic sheet stretched to 615% of its initial length. The top frame shows the sheet (blue) and rollers (black). At the bottom, we plot the reaction forces from the left side of the sheet.

We modeled the beam using St. Venant-Kirchhoff material ($E = 10\,\text{MPa}$, $\nu = 0.0$). The beam was also fixed along its depth so that it could only deflect vertically and horizontally (Figure 5, left). The method used to apply the load at the tip differed slightly between the two solvers. In FEBio, we applied the force as a nodal force split evenly between the four nodes at the top of the beam – meaning each node was subjected to 67.3375 N of force. PolyFEM integrates the load across the surface itself; therefore, we divided the applied force by the cross-sectional area (17 956.66 N) in order to make the two problems equivalent. We measured the amount of lateral deflection (displacement to the left) at the center of the top face, and the two solvers produced equivalent final results, with FEBio providing results faster (2 and 30 seconds, respectively). (Figure 5, right).

In the "upsetting of an elastic billet" simulation, we compressed to 60% of its initial height between two rigid surfaces a 1 mm × 1 mm × 0.1 mm elastic billet (meshed with 40 × 40 × 1 hexahedral elements, Figure 6, top). Using a quarter symmetry assumption, we can simulate only one-quarter of the elastic billet (Figure 6, top). This meant that the top and left sides of the billet were fixed perpendicular to those sides (modeled as rollers in Figure 6). We modeled the billet as a Mooney–Rivlin material ($C_1 = 1\,\text{MPa}$, $C_2 = 10\,\text{MPa}$, $K = 10\,\text{GPa}$). The rigid surface was modeled as an obstacle in PolyFEM and a rigid wall within FEBio. The outcome measure of this simulation is the maximum lateral displacement of the elastic billet. In this example, the solvers agreed on the lateral displacement of the top of the billet to the thousandths of a millimeter (0.781 mm, Figure 6 bottom). The simulation was significantly faster when conducted in FEBio than in PolyFEM, with a convergence time of 3 seconds and 56 seconds, respectively.

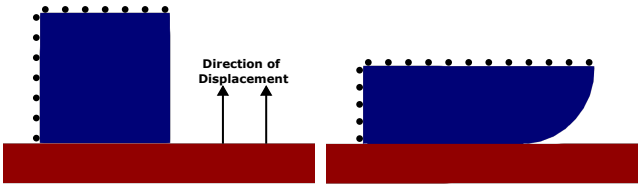



The cantilever beam in its undeformed (black vertical line) and deformed (blue curve) position. The red dashed arrow shows the force of 269.35 N applied to the top face in the $z$ direction. The force is enough to deflect the beam laterally by 8 m.




Plot showing that the displacement curves from the two solvers are nearly identical. The slight deviation along the beginning of the curve can likely be explained by the different methods of applying the tip load.

**Figure 5:** Cantilever comparison between FEBio and PolyFEM.




Elastic billet at the beginning of the simulation. Black circles represent rollers as the elastic billet was constrained perpendicular to those axes.

Elastic billet fully compressed.

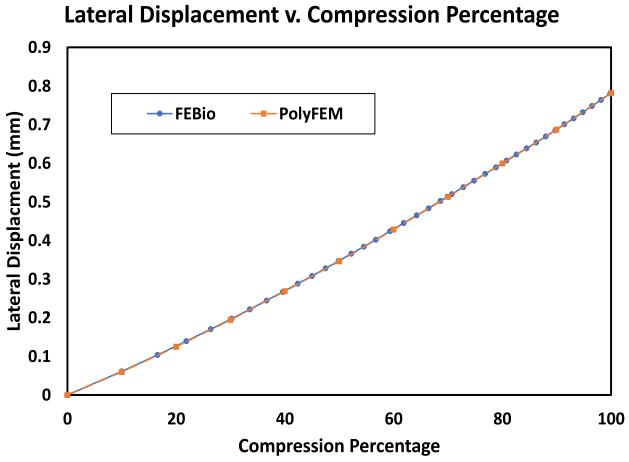



**Figure 6:** Elastic billet Figure displacement caused by the compression of two rigid surfaces.

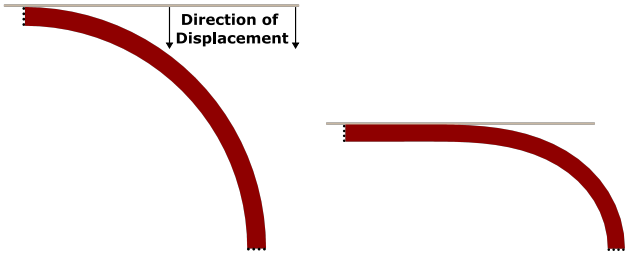



**Figure 7:** Model of a deformable pipe that was crushed between two rigid objects. The left image showed the pipe and rigid wall in the rest configuration while the image on the right showed the pipe which has been compressed to ≈ 46% its initial height

The final simulation consists of crushing a pipe ($r_o$ = 114.3 mm, $r_i$ = 105.43 mm, $t$ = 25.4 mm) by a rigid body. The pipe was modeled as a St. Venant-Kirchhoff material ($E$ = 185.86 GPa, $\nu$ = 0.299 72 ) and meshed it using FEBio (Slices = 24 , Segments = 4 , Stacks = 1 ). As in the previous simulation, the rigid body was modeled as a rigid wall and obstacle in Febio and PolyFEM, respectively. Additionally, quarter symmetry was assumed. Figure 7 shows the pipe and obstacle in the rest and deformed configurations. We were unable to directly compare the outcome measures calculated in the FEBio paper because PolyFEM does not calculate reaction forces on rigid objects. Instead, we compared the shape change between the two solvers, which were identical. Once again the simulation ran much faster in FEBio than PolyFEM (2 and 60 seconds, respectively. Although we were unable to compare the reaction force of the rigid body, we were able to show that PolyFEM is capable of correctly predicting contact-driven simulations.

We excluded five simulations from the benchmark in [43] as they contain unsupported features in PolyFEM: two tests model viscoelastic material responses, one uses shell elements, and the final two use rigid bodies and/or shell as integral components.

### 4.3. Uniaxial Compression of Soft Materials

We developed an extremely simplified knee model to emphasize the differences in contact enforcement between the two solvers, particularly in the presence of large deformations, soft materials, and complex contact scenarios.

Our goal was to show complications which appear in more complex simulations in the simplest manner possible. Therefore, we modeled the knee using three simple components: a triangular prism representing a meniscus, which was compressed between a sphere representing the femur and a rectangular prism representing the tibial plateau. We only allow the meniscus to displace downwards ($-y$ direction) and we fixed it in the other two directions on the left side ($-x$ direction). We fixed the rectangular prism in all three directions on the bottom face ($-y$ direction). Figure 8 top row shows the setup for both solvers. We displace the sphere downward, compressing the meniscus between itself and the rectangular prism until the meniscus experienced an average axial strain of approximately 18%, which falls within the normal physiological range for the contact area [11, 72, 48]. Initially, all objects in the simulation were not in contact, as required by IPC. We modeled the femur ($E = 17\,000$ MPa, $\upsilon = 0.3$, $\rho = 1850\,\mathrm{kgm^{-3}}$), tibial plateau ($E = 17\,000$ MPa, $\upsilon = 0.3$, , $\rho = 1850\,\mathrm{kgm^{-3}}$), and meniscus ($E = 59$ MPa, $\upsilon = 0.49$, $\rho = 800\,\mathrm{kgm^{-3}}$) as Neohookean materials [14, 53] and we meshed all objects using tetrahedral elements with FTetWild's default settings [30].

We defined the contact in FEBio as sliding elastic contact with an auto penalty and two-pass contact enabled. We setup two contact interfaces: one between the femur and the meniscus, and the other between the meniscus and the tibial plateau. We used the meniscus as the primary surface in both contact pairs as it is more finely discretized. Because sliding elastic contact is non-symmetric, we ensured that symmetric stiffness was turned off in both the contact menus and the analysis step. We adjusted the contact search radius until the surfaces no longer passed through each other; this occurred around 0.1. Within the solver step, we set `max_ups = 0` to enable the full Newton's method. We also set `rhoi = 1`, to use the backward Euler's method as time integrator (the method naively used in PolyFEM). In PolyFEM we defined contact by enabling contact and disabling adaptive barrier stiffness, instead opting for a constant value ($1 \times 10^9$). We simulated for 1 s with constant time steps ($t = 0.005$ s) for both solvers using `dtforce` in FEBio and by defining the time steps in PolyFEM.

We compared the simulation outcomes at four time points ($t = 0.0$ s, $t = 0.36$ s, $t = 0.72$ s, and end of simulation) shown in the rows of Figure 8). We report the contact enforcement variable (contact pressure in FEBio, and contact force in PolyFEM) in Figure 9. We note that FEBio failed to complete the simulation and therefore the final comparison is between $t = 0.95$ s for FEBio and $t = 1.0$ s for PolyFEM. We normalized the contact enforcement forces with respect to their maximum value to ease a direct comparison as the scale of the raw values are orders of magnitude different.

While the results shown in Figure 8 look very similar, there are significant differences in the forces used to resist intersections of bodies. We selected a group of five vertices (highlighted in the top row of Figure 8) on the surface of the meniscus within the area of contact of the femur. From

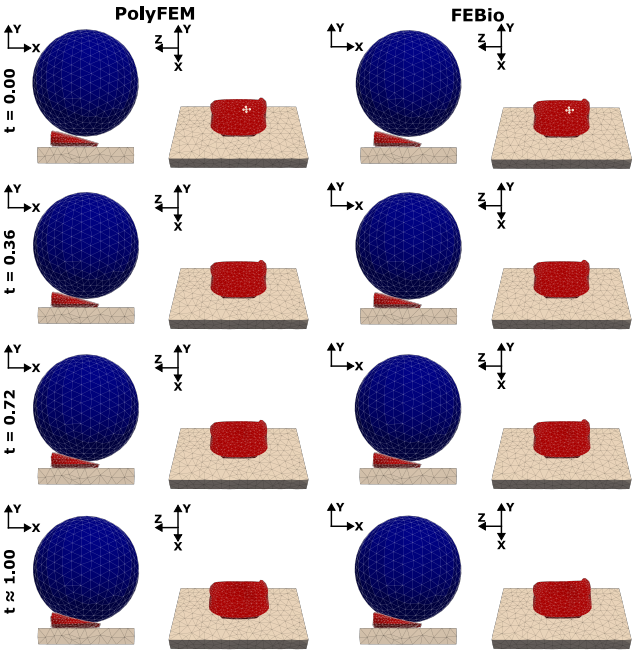



**Figure 8:** Images of the simulation output at four different time points, represented by the rows. The first two columns display the output from PolyFEM from two different views, while the third and fourth columns show the output from FEBio. To show the contact surface of the meniscus, the second row hides the femur (blue sphere). The top row corresponds to $t = 0.00$ s, the second row to $t = 0.36$ s, the third row to $t = 0.72$ s, and the fourth row represents the end of the simulation. Within the top row there are five points on the surface marked by the black dots. These points are used to calculate the average contact force in figure 9. FEBio encountered failure at $t = 0.95$ seconds, attributed to a negative Jacobian induced by contact. Images in left column were generated in PolyFEM and those in the right were generated using FEBio. Each set of geometries were visualized in Paraview.

these, we recorded the contact force for all time steps in both solvers and took the average. Figure 9 shows the contact pressure (FEbio) and contact force (PolyFEM) normalized to their respective maximum values. We chose these five points on the surface such that we can emphasize the difference in oscillations due to applications of the contact constraints between the two codes[2]. As the simulation compresses the meniscus, both contact forces increase in a similar fashion. However, around $t = 0.72$ s, the FEBio simulation becomes increasingly unstable, shown by the significant oscillations in the pressure. This is shown in the simulation by the unconstrained end of the meniscus moving from left to right on the $z$-axis in response to the oscillations of the contact pressure.

In this simulation, as with some other problems in this study, there may well be a set of parameters or mesh configurations that could yield better results in FEBio. However, this example underscores the fact that even in simple problems

[2]Comparing integrated contact forces over the surfaces of the meniscus and tibial plateau is currently not supported in PolyFEM.

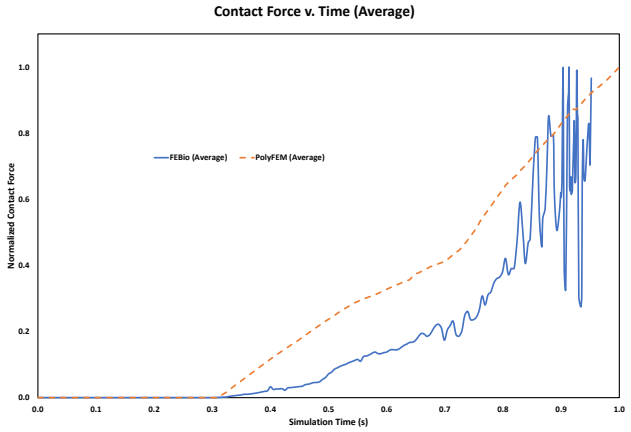



**Figure 9:** Chart of the average contact force for five points on the surface of meniscus normalized to their respective maximum value. The top row of figure 8 denotes the points used to calculate these plots. As compression of the meniscus increases the force required to resist intersection increases as a similar rate for both solvers. Around $t = 0.72\,\mathrm{s}$ the FEBio simulation becomes increasingly unstable and, due to the contact creates non-realistic side-to-side ($z$-axis) motion of the unconstrained end of the meniscus.

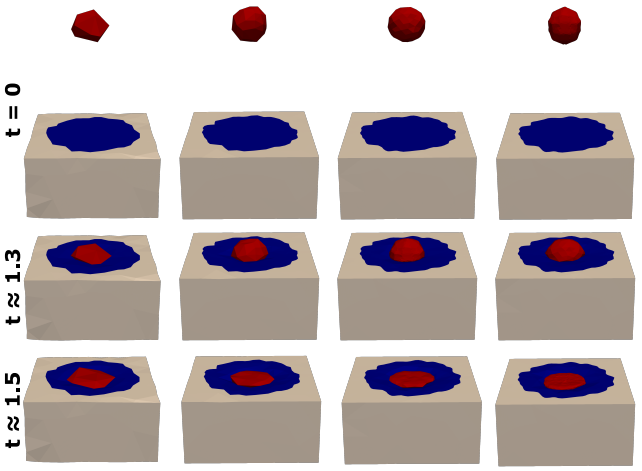



**Figure 10:** Steps of the simulation of a high energy collision between three deformable bodies (planet: red, jello: blue, and mold: off-white). Each column represents the same single simulation of the planet falling onto the jello at each of the discretization levels. For this simulation, the planet and jello had moduli of 10 kPa and 100 kPa, respectively. Each row represented a different time point of the simulation. The first row shows the initial configuration of the simulation, the second shows the first contact between the planet and jello ($t \approx 1.3$), and the third shows when the planet has compressed the most ($t \approx 1.5$). Images in the figure were generated using PolyFEM visualized in Paraview.

involving basic geometries, errors can occur. While we acknowledge that these errors can often guide model creators towards potential issues with geometries or contacts, there are situations where, despite parameter tuning and geometric adjustments, successful models couldn't be generated. It's essential to remember that any model generated in any solver should be appropriately validated by the model's creator before its use in any capacity, a point this study does not claim to address.

### 4.4. Planet-Fall

We developed a novel dataset to emphasize the differences between PolyFEM and FEBio, in particular in the presence of contact, large deformations, and soft materials. This specific simulation was not based on any biological phenomena, and instead aimed to represent potential high energy collisions in biomechanical systems is as simple of a model as possible. Every simulation in this dataset is composed of: a sphere-like (planet) modeled as a single material, and a rectangular prism consisting of two different materials, a relatively soft "inner" material (jello), which was encased on all sides except for the top surface by a relatively stiff outer material (mold). The three materials were all modeled as Neohookean. The bottom face of the mold was fixed in all three directions and all geometries were subject to gravity ($9.81\,\mathrm{ms}^{-2}$), which caused the planet to fall onto the top face of the jello (Figure 10). While high-energy contact between very soft materials is a smaller field of biomechanics, existing solvers struggle to simulate them properly. PolyFEM, for most users, will not replace existing solvers, but for the users that need to simulate contact like this example problem PolyFEM will be a valuable tool.

We repeated this setup 64 times using 4 different mesh densities to mimic the normal iterative process of FE modeling and variations of moduli between the 3 materials (Table 2). We meshed the planet with 20, 203, 2002, and 20 164 tetrahedral elements while the mold and jello were meshed with a combined 1958, 3951, 8034, and 16 045 tetrahedral elements. Each discretization level was paired with the equal mesh density of the other object in the simulation (e.g., the 20 tet planet was paired with the 1958 tet mold and jello). For each discretization level, we varied the material properties of the planet and the jello between 10 kPa and 10 000 kPa; while we kept the modulus of the mold's material constant at 100 GPa. Similarly, we fixed the Poisson's ratio and density of all objects ($\nu = 0.40$, $\rho = 1000\,\mathrm{kgm}^{-3}$).

We ran the simulation from a start time of 0 s to an end time of 2 s using $dt$ =0.1 s. FEBio allows for adaptive time stepping, which we enabled to utilize the aggressive cutback option. In PolyFEM, we simply enabled contact and solved for the prescribed $dt$ of 0.1 s. We recognize that differences in time stepping may result in slightly different solutions; however, the goal of this set of simulations was simply to determine if the two solvers could provide solutions for these test cases since the previous sections already provided verification. In the FEBio simulation, we specified the same contact parameters as stated above in section 4.3, sliding elastic contact using auto-penalty, backwards Euler time integration, and the full Newton's method.

**Table 2**
Runtimes of simulations in seconds. Each table corresponds to a mesh resolution; the rows and columns are the modulus of elasticity of the jello and planet, respectively. The entries in the table are of the form PolyFEM runtime (FEBio runtime); for instance, 13 (5) means that PolyFEM took 13 seconds while FEBio only took 5. We use NC when the solver fails to reach a solution.

| J\ P | $10^4$ Pa | $10^5$ Pa | $10^6$ Pa | $10^7$ Pa |
|---|---|---|---|---|
| $10^4$ Pa | 69 (NC) | 28 (NC) | 17 (NC) | 17 (NC) |
| $10^5$ Pa | 30 (NC) | 19 (NC) | 13 (45) | 13 (NC) |
| $10^6$ Pa | 21 (NC) | 17 (12) | 15 (4) | 11 (9) |
| $10^7$ Pa | 20 (NC) | 17 (NC) | 12 (3) | 14 (7) |

Planet (20 tet) – Jello (1958 tet)

| J\ P | $10^4$ Pa | $10^5$ Pa | $10^6$ Pa | $10^7$ Pa |
|---|---|---|---|---|
| $10^4$ Pa | 162 (NC) | 81 (NC) | 57 (NC) | 51 (NC) |
| $10^5$ Pa | 108 (NC) | 49 (115) | 43 (22) | 36 (NC) |
| $10^6$ Pa | 45 (NC) | 28 (38) | 24 (7) | 25 (18) |
| $10^7$ Pa | 38 (NC) | 30 (349) | 24 (6) | 21 (13) |

Planet (203 tet) – Jello (3951 tet)

| J\ P | $10^4$ Pa | $10^5$ Pa | $10^6$ Pa | $10^7$ Pa |
|---|---|---|---|---|
| $10^4$ Pa | 379 (NC) | 212 (NC) | 109 (NC) | 87 (NC) |
| $10^5$ Pa | 241 (NC) | 128 (2205) | 69 (90) | 66 (233) |
| $10^6$ Pa | 132 (NC) | 76 (NC) | 47 (25) | 43 (51) |
| $10^7$ Pa | 124 (NC) | 76 (NC) | 54 (45) | 47 (41) |

Planet (2002 tet) – Jello (8034 tet)

| J\ P | $10^4$ Pa | $10^5$ Pa | $10^6$ Pa | $10^7$ Pa |
|---|---|---|---|---|
| $10^4$ Pa | 2233 (NC) | 1072 (NC) | 356 (NC) | 220 (NC) |
| $10^5$ Pa | 1306 (NC) | 612 (NC) | 283 (NC) | 185 (NC) |
| $10^6$ Pa | 689 (NC) | 342 (NC) | 201 (255) | 177 (141) |
| $10^7$ Pa | 492 (NC) | 246 (NC) | 178 (146) | 139 (176) |

Planet (20 164 tet) – Jello (16 045 tet)

We captured the runtimes (using the same computer as Section 4.2) of the simulations and report them in Table 2; we report the successful timing for all simulations, but it is important to note that this did not include the time required to setup the model and examine each simulation for penetration. Additionally, we acknowledge that there exist configurations of contact parameters that produce significantly faster results. However, utilizing these methods necessitated more case-by-case tuning to achieve a converged solution. In contrast, employing the outlined contact parameters yielded results more reliably compared to using manually defined penalties and a quasi-Newton solver, despite requiring a significantly longer time to complete. Overall, it was found that most of the simulations ran faster in FEBio for the simulations where FEBio succeeded. However, it was unable to complete the entire benchmark, including the majority of the finest mesh density examples. All simulations that did not complete, failed due to the presence of negative Jacobians, either due to contact or elasticity. Meanwhile, all of the simulations for PolyFEM completed successfully.

To provide a quantitative idea of the setup complexity, we prepared all simulations on the same computer (2017 iMac Pro, which was used to establish the runtimes). The PolyFEM simulations were prepared using the JSON file format and the FEBio simulations were prepared in FEBio Studio (version 1.8). We note that the authors were well-versed in using both of the software packages and are familiar with the process of setting up this simulation; that is, this was not a blind test. We prepared the simulations a total of five times each in alternating order (Day 1: PolyFEM, then FEBio, Day 2: FEBio, then PolyFEM, etc) on five subsequent days. We average the processing time excluding the fastest and slowest preparation time. The setup times for each of the two software packages were similar for a familiar user (219 s for PolyFEM, 214 s for FEBio).

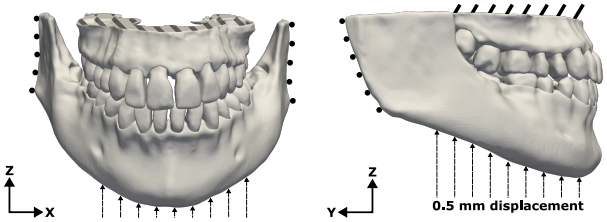



**Figure 11:** Required boundary conditions applied to the human jaw to setup each simulation. Dashes represented boundaries that were fixed in all directions and rollers represented boundaries fixed perpendicular to them. The upper jaw was fixed in all directions, and the lower jaw was fixed in $x$ and $y$ directions and displaced 0.5 mm in the positive $z$-direction.

However, the biggest time save in PolyFEM (excluding the failures and subsequent parameter tuning) is when switching between the different meshes. PolyFEM only requires changing the file path to the geometry, as the boundary conditions and forces will still be applied properly. In contrast, in FEBio, the boundary conditions and body forces need to be manually recalculated and reapplied the first time that the mesh is changed.

## 4.5. Hip-and-Jaw

In the following two sections we conducted biomechanical simulations of biting and pseudo-stance using PolyFEM and FEBio. To perform the simulations, we utilized patient-specific finite element (FE) models of the hip and jaw, obtained from the publicly available LibHip [49] and Open-Full-Jaw [25] repositories. The applied boundary conditions are illustrated in Figure 11 and Figure 13. Both simulations were simulated statically by setting the solver to static in FEBio, "ignore_inertia" set to true in PolyFEM.

In the jaw simulation, we fixed the upper region of the maxilla mesh in all three directions, while the lower part of the mandible mesh was displaced by 0.5 mm in the positive $z$-direction. The mandible's side nodes were fixed in the $x$ and $y$ directions. Each anatomical structures in both models were assumed homogeneous and modeled as Neohookean materials, with the teeth having $E = 2\,\text{GPa}$ and $\nu = 0.3$, the periodontal ligament having $E = 68.9\,\text{MPa}$ and $\nu = 0.45$, and the jaw bone having $E = 1.5\,\text{GPa}$ and $\nu = 0.30$.

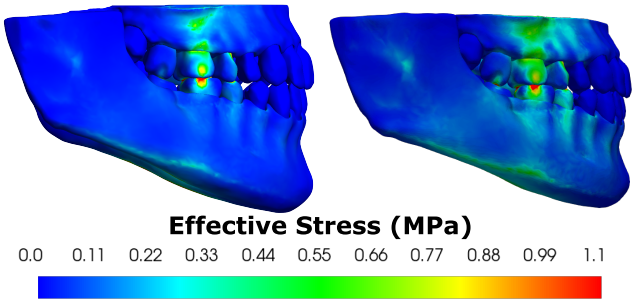



**Figure 12:** The outputs from the jaw simulation from both of the FEBio (left) and PolyFEM (right). The two solvers produced very similar stress distributions across the surfaces.

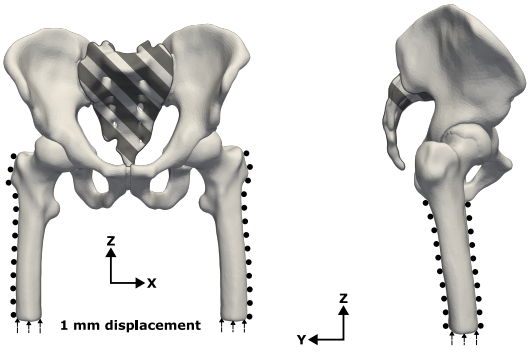



**Figure 13:** Required boundary conditions applied to the human hip (bottom) to setup each simulation. Dashes represented boundaries that were fixed in all directions and rollers represented boundaries fixed perpendicular to them. The sacrum was fixed in all three directions, and the femurs were both fixed in the $x$ and $y$ directions. A displacement of 1 mm was applied to the distal end of the femur in the positive $z$-direction.

In the hip simulation, we restricted the pelvic girdle by fixing the sacrum's displacement and rotation in the $x$, $y$, and $z$ directions. We also displaced the distal ends of the two femurs by 1 mm in the positive $z$-direction while restricting the rest of the femur in the $x$ and $y$ directions. The components of the model were simulated as Neohookean materials, with the hip bone having $E = 17$ GPa and $\nu = 0.30$, and the hip cartilage having $E = 12$ MPa and $\nu = 0.45$.

Due to the different contact models utilized in FEBio and PolyFEM, some simulation setups could not be replicated in both solvers. For instance, FEBio necessitated an initial penetration between contact surfaces for accurate contact detection, while PolyFEM did not require such an initial step. Therefore, to ensure simulation convergence, the distance between contact surfaces, such as the separation between two sliding cartilages in the hip joint, must be tailored to the specific solver used. If the contact model in the solver is not taken into consideration the likelihood of the simulation failing to converge raises significantly. For example, in the biting scenario, if the prescribed displacement is replaced with a pressure load, the simulation fails to converge in FEBio due to inverted elements. This could have been caused by a few different issues like incorrect contact parameters, and

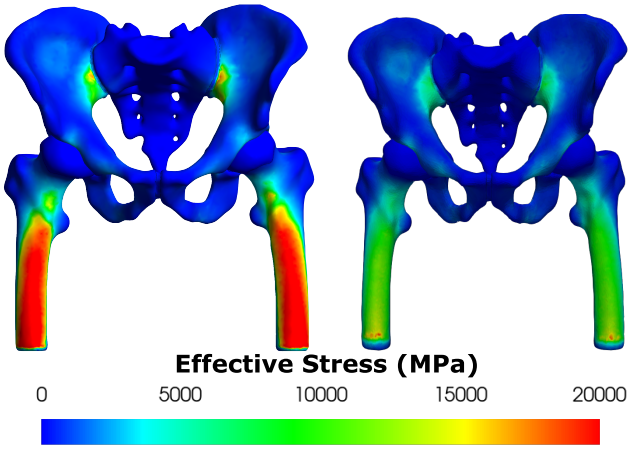



**Figure 14:** The outputs from the hip simulation from both of the FEBio (left) and PolyFEM (right). The two solvers produced very similar stress distributions across the surfaces. The only major difference was in the stress distributions of this simulations, this is most likely due to the differences in how the boundary conditions were applied to the models. The model created in FEBio also used rigid regions of bone to drive the displacement, which may have also contributed to differences in stress distribution.

despite our efforts to modify the applied load magnitude and adjust various contact parameters, the simulations failed in all our attempts. Some groups have found success using a two step analysis, starting with an initial small displacement step which is then followed by the pressure step for contact in a simulation driven by pressure boundary conditions. This may have helped drive this simulation to convergence however, PolyFEM's contact formulation does not require this additional tuning which can only be discovered after a simulation fails. These findings suggest that simulations involving complex geometries, soft tissues, contacts, and complex loading conditions may require significant parameter-tuning procedures to achieve successful results. Contrarily, PolyFEM was able to run the simulation with the pressure load without issues, and did not require parameter tuning.

To improve the accuracy of the simulations and avoid element-locking effects, we increased the order of the volume mesh elements. FEBio's user interface includes a tool that allows for the conversion of different element types into one another. In our case, we converted *Tet4* (linear tetrahedral) elements into *Tet10* (quadradic tetrahedral) elements, which resulted in an increase in the number of nodes in the jaw model by nearly 700%. However, due to the excessive memory required for such a large simulation, we were unable to execute the simulations using FEBio on the same computer that was used in section 4.2 and 4.4. Using PolyFEM, we benefited from the adaptive $p$-refinement feature, which allowed us to selectively increase the order of the basis functions used in specific domains while employing linear basis functions for the remaining domains. Using this method, we were able to perform the simulations on the same machine, and the convergence time was approximately 17

and 42 minutes in FEBio and PolyFEM, respectively. The hip simulation took significantly more time to complete in PolyFEM than FEBio (4 hours and 5 minutes, respectively). This is likely due to the differences in the contact and methods for applying the boundary conditions. Future releases of PolyFEM will need to focus on addressing the significantly longer simulation time, however, as we have noted before there was no tuning of the models required in PolyFEM while there was a significant amount of tuning for FEBio during the initial model development like adjusting penalty factors and adjusting intersection amounts. Additionally, we could simulate a different hip geometry without needing to change any of the boundary conditions.

## 5. Conclusions

This study demonstrated that PolyFEM produced results matching those from FEBio, and by extension some other FE solvers for previously published simulations based on solid, hyperelastic materials. This provides important verification of the solutions provided by PolyFEM. Further, this study demonstrated that PolyFEM offers solutions to problems that are challenging for other solvers, such as contact, soft materials , and/or extreme deformations. Even though for the vast majority of biomechanics simulations existing solvers are sufficient, there exists a subset of problems that were previous extremely difficult to simulate. PolyFEM targets these simulations. Finally, this study demonstrated the utility of PolyFEM in solving patient-specific models in biomechanics. Thus, this alternative solver is very suitable for solving many problems in biomechanics where geometric nonlinearities are common.

We believe IPC-based solvers are an ideal fit for biomechanical simulation, despite their current restricted scope, and our work provides guidelines and benchmarks to support the development and research of these techniques for biomechanical purposes. We are excited by the prospect of having the IPC-based solvers in biomechanics, as we believe they could lead to a massive reduction in human effort and open the door to a larger use of simulation for designing and understanding biomechanical systems.

It is important to note that at this stage of development, PolyFEM lacks many of the features available for other solvers. These include a user interface (PolyFEM uses a JSON setup file and Paraview for post-processing), a wide array of materials, shell and rod elements, a rigid body solver, tied-contact, a multi-physics platform, and optimization for parallel performance. However, there are plans to implement many of these features, which would help PolyFEM realize its high potential for biomechanical simulation due to its improved automation and robustness. It should also be noted that one limitation of using a barrier potential for contact is that the simulation cannot have interpenetrating surfaces in its initial configuration. This will need to be considered when creating meshes from segmented medical images. The upside of this limitation is the higher robustness and the guarantee that there will be no penetrations in all timesteps. Moreover, there are parameters that can be adjusted that can lead to improved performance for challenging simulations, including the barrier stiffness and d-hat parameters. In some cases in this manuscript we adjusted these parameters such that a solutions was reached more efficiently however, using the automatic formulation of barrier stiffness also produced similar results at the cost of computational time. At the time of writing this manuscript, those parameters have not been fully optimized, so it is anticipated that the difference in runtimes between the two solvers will improve with further development.

While PolyFEM is early in its development, it currently provides verified solutions for hyperelastic materials that are consistent with FEBio, and it is capable of simulating challenging problems in biomechanics where other solvers fail. It is also open-source and publicly available. Future work will aim to implement many of the aforementioned features to provide more options for the biomechanics community to implement it as another tool in their workflows.

## ACKNOWLEDGEMENTS

We acknowledge funding from NSF 2053851, NSF 1835712, NSF 1908767, NIH R01HD097187, NIH R01-HD083383-06, NSERC DGECR-2021-00461, RGPIN 2021-0370, Horizon2020 MSCA grant No.764644, and NIH R01DK133328.

## References

[1] Abaqus, 2006. Abaqus Verification Manual. Verification Manual 6.6. Dassault Systems. Vélizy-Villacoublay, France.

[2] Ammar, H.H., Ngan, P., Crout, R.J., Mucino, V.H., Mukdadi, O.M., 2011. Three-dimensional modeling and finite element analysis in treatment planning for orthodontic tooth movement. American Journal of Orthodontics and Dentofacial Orthopedics 139, e59–e71.

[3] Anderson, A.E., Ellis, B.J., Weiss, J.A., 2007. Verification, validation and sensitivity studies in computational biomechanics. Computer Methods in Biomechanics and Biomedical Engineering 10, 171–184. doi:10.1080/10255840601160484.

[4] Ansys, 2013. ANSYS Mechanical APDL Verification Manual. ANSYS Verification Manual 15. ANSYS inc.. Cannonsburg, PA.

[5] Ateshian, G., Maas, S.A., Rossherron, M., 2022. FEBio software: Testsuite. https://github.com/febiosoftware/TestSuite.

[6] Babuska, I., 1973. The Finite Element Method with Penalty. Mathematics of Computation 27, 221. doi:10.2307/2005611.

[7] Bathe, K., 1996. Finite Element Procedures. 2nd ed., Prentice-Hall, Watertown, MA.

[8] Bog, T., Zander, N., Kollmannsberger, S., Rank, E., 2015. Normal contact with high order finite elements and a fictitious contact material. Computers & Mathematics with Applications 70, 1370–1390. doi:10.1016/j.camwa.2015.04.020.

[9] Burman, E., Claus, S., Hansbo, P., Larson, M.G., Massing, A., 2015. CutFEM: Discretizing geometry and partial differential equations. International Journal for Numerical Methods in Engineering 104, 472–501. doi:10.1002/nme.4823.

[10] Chethan, K., Zuber, M., Shenoy, S., Kini, C.R., et al., 2019. Static structural analysis of different stem designs used in total hip arthroplasty using finite element method. Heliyon 5, e01767.

[11] Chia, H.N., Hull, M.L., . Compressive moduli of the human medial meniscus in the axial and radial directions at equilibrium and at a physiological strain rate 26, 951–956. URL: https://onlinelibrary.wiley.com/doi/10.1002/jor.20573, doi:10.1002/jor.20573.

[12] Claus, S., Kerfriden, P., Moshfeghifar, F., Darkner, S., Erleben, K., Wong, C., 2021. Contact modeling from images using cut finite element solvers. Advanced Modeling and Simulation in Engineering Sciences 8, 13. doi:10.1186/s40323-021-00197-2.

[13] Damsgaard, M., Rasmussen, J., Christensen, S.T., Surma, E., de Zee, M., 2006. Analysis of musculoskeletal systems in the anybody modeling system. Simulation Modelling Practice and Theory 14, 1100–1111. doi:https://doi.org/10.1016/j.simpat.2006.09.001. sIMS 2004.

[14] Du, M., Sun, J., Liu, Y., Wang, Y., Yan, S., Zeng, J., Zhang, K., . Tibiofemoral contact force distribution of knee before and after total knee arthroplasty: Combined finite element and gait analysis 14, 1836–1845. URL: https://onlinelibrary.wiley.com/doi/10.1111/os.13361, doi:10.1111/os.13361.

[15] Dubov, A., Kim, S.Y.R., Shah, S., Schemitsch, E.H., Zdero, R., Bougherara, H., 2011. The biomechanics of plate repair of periprosthetic femur fractures near the tip of a total hip implant: the effect of cable-screw position. Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine 225, 857–865. doi:10.1177/0954411911410642.

[16] Elshazly, T.M., Bourauel, C., Aldesoki, M., Ghoneima, A., Abuzayda, M., Talaat, W., Talaat, S., Keilig, L., 2022. Computer-aided finite element model for biomechanical analysis of orthodontic aligners. Clinical Oral Investigations 27, 115–124. doi:10.1007/s00784-022-04692-7.

[17] Erdemir, A., Guess, T.M., Halloran, J., Tadepalli, S.C., Morrison, T.M., 2012. Considerations for reporting finite element analysis studies in biomechanics. Journal of Biomechanics 45, 625–633. doi:10.1016/j.jbiomech.2011.11.038.

[18] Erdemir, A., Mulugeta, L., Ku, J.P., Drach, A., Horner, M., Morrison, T.M., Peng, G.C.Y., Vadigepalli, R., Lytton, W.W., Myers, J.G., 2020. Credible practice of modeling and simulation in healthcare: ten rules from a multidisciplinary perspective. Journal of Translational Medicine 18, 369. doi:10.1186/s12967-020-02540-4.

[19] Erleben, K., 2018. Methodology for Assessing Mesh-Based Contact Point Methods. ACM Transactions on Graphics 37, 1–30. doi:10.1145/3096239.

[20] Fang, G., Lin, Y., Wu, J., Cui, W., Zhang, S., Guo, L., Sang, H., Huang, W., 2020. Biomechanical Comparison of Stand-Alone and Bilateral Pedicle Screw Fixation for Oblique Lumbar Interbody Fusion Surgery—A Finite Element Analysis. World Neurosurgery 141, e204–e212. doi:10.1016/j.wneu.2020.05.245.

[21] Faure, F., Duriez, C., Delingette, H., Allard, J., Gilles, B., Marchesseau, S., Talbot, H., Courtecuisse, H., Bousquet, G., Peterlik, I., Cotin, S., 2012. SOFA: A Multi-Model Framework for Interactive Physical Simulation, in: Payan, Y. (Ed.), Soft Tissue Biomechanical Modeling for Computer Assisted Surgery. Springer. volume 11 of *Studies in Mechanobiology, Tissue Engineering and Biomaterials*, pp. 283–321. doi:10.1007/8415_2012_125.

[22] Fernandes, F.A.O., Alves de Sousa, R.J., Ptak, M., 2018. Validation of YEAHM, in: Head Injury Simulation in Road Traffic Accidents. Springer International Publishing, Cham. SpringerBriefs in Applied Sciences and Technology, pp. 41–58. doi:10.1007/978-3-319-89926-8.

[23] Finley, S.M., Brodke, D.S., Spina, N.T., DeDen, C.A., Ellis, B.J., 2018. FEBio finite element models of the human lumbar spine. Computer Methods in Biomechanics and Biomedical Engineering 21, 444–452. doi:10.1080/10255842.2018.1478967.

[24] Geng, J.P., Tan, K.B., Liu, G.R., 2001. Application of finite element analysis in implant dentistry: a review of the literature. The Journal of prosthetic dentistry 85, 585–598.

[25] Gholamalizadeh, T., Moshfeghifar, F., Ferguson, Z., Schneider, T., Panozzo, D., Darkner, S., Makaremi, M., Chan, F., Søndergaard, P.L., Erleben, K., 2022. Open-Full-Jaw: An open-access dataset and pipeline for finite element models of human jaw. Computer Methods and Programs in Biomedicine 224, 107009. doi:10.1016/j.cmpb.2022.107009.

[26] Guo, H., Nickel, J.C., Iwasaki, L.R., Spilker, R.L., 2012. An augmented lagrangian method for sliding contact of soft tissue. Journal of Biomechanical Engineering .

[27] Guo, H., Spilker, R.L., 2011. Biphasic finite element modeling of hydrated soft tissue contact using an augmented lagrangian method. Journal of biomechanical engineering 133.

[28] Henninger, H.B., Reese, S.P., Anderson, A.E., Weiss, J.A., 2010. Validation of computational models in biomechanics. Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine 224, 801–812. doi:10.1243/09544119JEIM649.

[29] Hu, P., Wu, T., Wang, H., Qi, X., Yao, J., Cheng, X., Chen, W., Zhang, Y., 2019. Biomechanical Comparison of Three Internal Fixation Techniques for Stabilizing Posterior Pelvic Ring Disruption: A 3D Finite Element Analysis. Orthopaedic Surgery 11, 195–203. doi:10.1111/os.12431.

[30] Hu, Y., Schneider, T., Wang, B., Zorin, D., Panozzo, D., 2020. Fast tetrahedral meshing in the wild. ACM Trans. Graph. 39. URL: https://doi.org/10.1145/3386569.3392385, doi:10.1145/3386569.3392385.

[31] Hughes, T.J., Taylor, R.L., Sackman, J.L., Curnier, A., Kanoknukulchai, W., 1976. A finite element method for a class of contact-impact problems. Computer Methods in Applied Mechanics and Engineering 8, 249–276. doi:10.1016/0045-7825(76)90018-9.

[32] Humphrey, J.D., Rajagopal, K., 2002. A constrained mixture model for growth and remodeling of soft tissues. Mathematical models and methods in applied sciences 12, 407–430.

[33] Jiang, T., Wu, R.Y., Wang, J.K., Wang, H.H., Tang, G.H., 2020. Clear aligners for maxillary anterior en masse retraction: a 3d finite element study. Scientific reports 10, 10156.

[34] Kamensky, D., Xu, F., Lee, C.H., Yan, J., Bazilevs, Y., Hsu, M.C., . A contact formulation based on a volumetric potential: Application to isogeometric simulations of atrioventricular valves 330, 522–546. URL: https://linkinghub.elsevier.com/retrieve/pii/S0045782517307120, doi:10.1016/j.cma.2017.11.007.

[35] Kane, C., Marsden, J.E., Ortiz, M., West, M., 2000. Variational integrators and the newmark algorithm for conservative and dissipative mechanical systems. Int. J. for Numer. Meth. in Eng. 49.

[36] Kim, Y.S., Kang, K.T., Son, J., Kwon, O.R., Choi, Y.J., Jo, S.B., Choi, Y.W., Koh, Y.G., 2015. Graft Extrusion Related to the Position of Allograft in Lateral Meniscal Allograft Transplantation: Biomechanical Comparison Between Parapatellar and Transpatellar Approaches Using Finite Element Analysis. Arthroscopy: The Journal of Arthroscopic & Related Surgery 31, 2380–2391.e2. doi:10.1016/j.arthro.2015.06.030.

[37] Laursen, T.A., 2002. Computational Contact and Impact Mechanics. Springer.

[38] Lew, S., Wolters, C., Dierkes, T., Röer, C., MacLeod, R., 2009. Accuracy and run-time comparison for different potential approaches and iterative solvers in finite element method based EEG source analysis. Applied Numerical Mathematics 59, 1970–1988. doi:10.1016/j.apnum.2009.02.006.

[39] Li, J., 2021. Development and validation of a finite-element musculoskeletal model incorporating a deformable contact model of the hip joint during gait. Journal of the Mechanical Behavior of Biomedical Materials 113, 104136. doi:10.1016/j.jmbbm.2020.104136.

[40] Li, M., Ferguson, Z., Schneider, T., Langlois, T., Zorin, D., Panozzo, D., Jiang, C., Kaufman, D.M., 2020. Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. ACM Transactions on Graphics 39. doi:10.1145/3386569.3392425.

[41] Logg, A., Wells, G.N., 2010. Dolfin: Automated finite element computing. ACM Trans. Math. Softw. 37. doi:10.1145/1731022.1731030.

[42] Luo, J., Chen, L., Fenner, D.E., Ashton-Miller, J.A., DeLancey, J.O., 2015. A multi-compartment 3-D finite element model of rectocele and its interaction with cystocele. Journal of Biomechanics 48, 1580–1586. doi:10.1016/j.jbiomech.2015.02.041.

[43] Maas, S.A., Ellis, B.J., Ateshian, G.A., Weiss, J.A., 2012. FEBio: Finite elements for biomechanics. J Biomech Eng 134.

[44] Mazier, A., Bilger, A., Forte, A.E., Peterlik, I., Hale, J.S., Bordas, S.P.A., 2022. Inverse deformation analysis: an experimental and numerical assessment using the fenics project. Engineering with Computers 38, 4099–4113. doi:10.1007/s00366-021-01597-z.

[45] Ménager, E., Schegg, P., Khairallah, E., Marchal, D., Dequidt, J., Preux, P., Duriez, C., 2022. SofaGym: An open platform for Reinforcement Learning based on Soft Robot simulations. Soft Robotics .

[46] Meng, Q., Jin, Z., Fisher, J., Wilcox, R., 2013. Comparison between FEBio and Abaqus for biphasic contact problems. Proceedings of the Institution of Mechanical Engineers. Part H, Journal of engineering in medicine 227, 1009–1019. doi:10.1177/0954411913483537.

[47] Mengoni, M., 2021. Biomechanical modelling of the facet joints: a review of methods and validation processes in finite element analysis. Biomechanics and Modeling in Mechanobiology 20, 389–401. doi:10.1007/s10237-020-01403-7.

[48] Morejon, A., Norberg, C.D., De Rosa, M., Best, T.M., Jackson, A.R., Travascio, F., . Compressive properties and hydraulic permeability of human meniscus: Relationships with tissue structure and composition 8, 622552. URL: https://www.frontiersin.org/articles/10.3389/fbioe.2020.622552/full, doi:10.3389/fbioe.2020.622552.

[49] Moshfeghifar, F., Gholamalizadeh, T., Ferguson, Z., Schneider, T., Nielsen, M.B., Panozzo, D., Darkner, S., Erleben, K., 2022. LibHip: An open-access hip joint model repository suitable for finite element method simulation. Computer Methods and Programs in Biomedicine 226, 107140. doi:10.1016/j.cmpb.2022.107140.

[50] Nocedal, J., Wright, S.J., 2006. Penalty and Augmented Lagrangian Methods, in: Numerical Optimization. Springer New York, New York, NY, pp. 497–528. doi:10.1007/978-0-387-40065-5_17.

[51] Oefner, C., Herrmann, S., Kebbach, M., Lange, H.E., Kluess, D., Woiczinski, M., 2021. Reporting checklist for verification and validation of finite element analysis in orthopedic and trauma biomechanics. Medical Engineering & Physics 92, 25–32. doi:10.1016/j.medengphy.2021.03.011.

[52] Peng, M.J.Q., Xu, H., Chen, H.Y., Lin, Z., Li, X., Shen, C., Lau, Y., He, E., Guo, Y., 2020. Biomechanical analysis for five fixation techniques of Pauwels-III fracture by finite element modeling. Computer Methods and Programs in Biomedicine 193, 105491. doi:10.1016/j.cmpb.2020.105491.

[53] Pfeiler, T.W., . Finite element nonlinear dynamic response analysis of the human knee joint. URL: https://trace.tennessee.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=6302&context=utk_gradthes.

[54] Pirmoradian, M., Naeeni, H.A., Firouzbakht, M., Toghraie, D., Darabi, R., et al., 2020. Finite element analysis and experimental evaluation on stress distribution and sensitivity of dental implants to assess optimum length and thread pitch. Computer methods and Programs in Biomedicine 187, 105258.

[55] Puso, M.A., Laursen, T.A., 2004. A mortar segment-to-segment contact method for large deformation solid mechanics. Computer Methods in Applied Mechanics and Engineering 193, 601–629. doi:10.1016/j.cma.2003.10.010.

[56] Ratajczak, M., Ptak, M., Chybowski, L., Gawdzińska, K., Będziński, R., 2019. Material and Structural Modeling Aspects of Brain Tissue Deformation under Dynamic Loads. Materials 12, 271. doi:10.3390/ma12020271.

[57] Routzong, M.R., Martin, L.C., Rostaminia, G., Abramowitch, S., 2021. Urethral support in female urinary continence part 2: a computational, biomechanical analysis of Valsalva. International Urogynecology Journal doi:10.1007/s00192-021-04694-1. publisher: International Urogynecology Journal.

[58] Routzong, M.R., Moalli, P.A., Maiti, S., De Vita, R., Abramowitch, S.D., 2019. Novel simulations to determine the impact of superficial perineal structures on vaginal delivery. Interface Focus 9, 20190011. doi:10.1098/rsfs.2019.0011.

[59] Sauer, R.A., De Lorenzis, L., . A computational contact formulation based on surface potentials 253, 369–395. URL: https://linkinghub.elsevier.com/retrieve/pii/S0045782512002769, doi:10.1016/j.cma.2012.09.002.

[60] Schneider, T., Dumas, J., Gao, X., Zorin, D., Panozzo, D., 2019. Polyfem. https://polyfem.github.io/.

[61] Seth, A., Hicks, J.L., Uchida, T.K., Habib, A., Dembia, C.L., Dunne, J.J., Ong, C.F., DeMers, M.S., Rajagopal, A., Millard, M., Hamner, S.R., Arnold, E.M., Yong, J.R., Lakshmikanth, S.K., Sherman, M.A., Ku, J.P., Delp, S.L., 2018. Opensim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement. PLOS Computational Biology 14, 1–20. doi:10.1371/journal.pcbi.1006223.

[62] Shu, L., Li, S., Sugita, N., 2020. Systematic review of computational modelling for biomechanics analysis of total knee replacement. Biosurface and Biotribology 6, 3–11.

[63] Simo, J.C., Wriggers, P., Taylor, R.L., 1985. A perturbed Lagrangian formulation for the finite element solution of contact problems. Computer Methods in Applied Mechanics and Engineering 50, 163–180. doi:10.1016/0045-7825(85)90088-X.

[64] Song, M., Sun, K., Li, Z., Zong, J., Tian, X., Ma, K., Wang, S., 2021. Stress distribution of different lumbar posterior pedicle screw insertion techniques: a combination study of finite element analysis and biomechanical test. Scientific Reports 11, 129–68. doi:10.1038/s41598-021-90686-6.

[65] Takhounts, E., Eppinger, R., Campbell, J.Q., Rabih, T., Power, E., Shook, L., 2003. On the Development of the SIMon Finite Element Head Model. Stapp Car Crash Journal 2003, 107–33.

[66] Takhounts, E.G., Ridella, S.A., Hasija, V., Tannous, R.E., Campbell, J.Q., Malone, D., Danelson, K., Stitzel, J., Rowson, S., Duma, S., 2008. Investigation of traumatic brain injuries using the next generation of simulated injury monitor (SIMon) finite element head model. Stapp Car Crash Journal 52. doi:10.4271/2008-22-0001.

[67] Taylor, R.L., 1980. Contact-impact Problems: Engineering report and user's manual. volume 1. The Administration.

[68] Van Staden, R., Guan, H., Loo, Y.C., 2006. Application of the finite element method in dental implant research. Computer methods in biomechanics and biomedical engineering 9, 257–270.

[69] Wang, B., Ferguson, Z., Schneider, T., Jiang, X., Attene, M., Panozzo, D., 2021. A large scale benchmark and an inclusion-based algorithm for continuous collision detection. ACM Trans. Graphic. 40.

[70] Wriggers, P., 1995. Finite element algorithms for contact problems. Archives of Computational Methods in Engineering 2, 1–49. doi:10.1007/BF02736195.

[71] Wriggers, P., Zavarise, G., 2007. A formulation for frictionless contact problems using a weak form introduced by Nitsche. Computational Mechanics 41, 407–420. doi:10.1007/s00466-007-0196-4.

[72] Yang, N., Canavan, P., Nayeb-Hashemi, H., Najafi, B., Vaziri, A., . Protocol for constructing subject-specific biomechanical models of knee joint 13, 589–603. URL: http://www.tandfonline.com/doi/abs/10.1080/10255840903389989, doi:10.1080/10255840903389989.

[73] Zavarise, G., De Lorenzis, L., 2009. The node-to-segment algorithm for 2D frictionless contact: Classical formulation and special cases. Computer Methods in Applied Mechanics and Engineering 198, 3428–3451. doi:10.1016/j.cma.2009.06.022.

[74] Zeng, Z., Cotin, S., Courtecuisse, H., 2022. Real-Time FE Simulation for Large-Scale Problems Using Precondition-Based Contact Resolution and Isolated DOFs Constraints. Computer Graphics Forum 41, 418–434. doi:10.1111/cgf.14563.

[75] Zhang, K., Li, L., Yang, L., Shi, J., Zhu, L., Liang, H., Wang, X., Yang, X., Jiang, Q., 2019. The biomechanical changes of load distribution with longitudinal tears of meniscal horns on knee joint: a finite element analysis. Journal of Orthopaedic Surgery and Research 14, 237. doi:10.1186/s13018-019-1255-1.

[76] Zhou, Z., Li, X., Kleiven, S., Shah, C.S., Hardy, W.N., 2018. A Reanalysis of Experimental Brain Strain Data: Implication for Finite Element Head Model Validation. Stapp Car Crash Journal 62, 293–318. doi:10.4271/2018-22-0007.

[77] Zimmerman, B.K., Ateshian, G.A., 2018. A Surface-to-Surface Finite Element Algorithm for Large Deformation Frictional Contact in febio. Journal of Biomechanical Engineering 140. URL: https://doi.org/10.1115/1.4040497, doi:10.1115/1.4040497. 081013.

[78] Zupancic Cepic, L., Frank, M., Reisinger, A., Pahr, D., Zechner, W., Schedle, A., 2022. Biomechanical finite element analysis of

short-implant-supported, 3-unit, fixed CAD/CAM prostheses in the posterior mandible. International Journal of Implant Dentistry 8, 8. doi:10.1186/s40729-022-00404-8.