

Comparing Algorithms for Soft Tissue Deformation: Accuracy Metrics and Benchmarks

Ron Alterovitz and Ken Goldberg
IEOR and EECS Departments, UC Berkeley
ron@ieor.berkeley.edu, goldberg@ieor.berkeley.edu

Abstract – There are a number of algorithms for interactively modeling deformable objects, such as soft human tissues. Because of the desire for real-time performance, approximations are made to quickly calculate deformations. This paper proposes accuracy metrics and benchmarks to allow a systematic comparison of simulation algorithms for deformable objects in 2D and 3D. We implement and compare the computed deformations of several algorithms based on the finite element method (FEM) that rely on the approximations of explicit time integration, mass lumping, or quasi-static motion. We also compare different material properties and show that some algorithms limit the breadth of materials that can be simulated.

I. INTRODUCTION

Many algorithms have been proposed for real-time simulation of deformable objects, each making its own tradeoff between speed and realism. In graphics applications, such as cloth animation, it is only necessary for the deformation to “look right.” The code provides tweakable parameters to an animator in order to produce the desired artistic effect. In such cases, numeric realism metrics are not relevant since visually pleasing results are the goal. However, for applications such as human surgery simulation, a more stringent and quantifiable level of realism is desired.

Among currently available approaches, the finite element method (FEM) yields the most realistic results because it is based on the equations of continuum mechanics. Unfortunately, execution speed is a major concern when running FEM simulations because of the large computational cost required to solve a system of equations for each simulation frame. Algorithms designed to solve these systems make varying levels of approximations in order to decrease computation time. To our knowledge, no one has experimentally compared the algorithms to determine the loss of realism that results from the approximations when simulating soft tissues.

We propose accuracy metrics for algorithms that calculate deformations in section III. These general metrics can be applied to any system that models deformable objects. Test cases are presented in section IV for use as benchmarks. FEM solver algorithms that we test are defined in section V and the results of our

experiments are presented in section VI.

II. RELATED WORK

Modeling deformable objects has a long history in mechanical engineering, including the development of FEM in the 1950’s [17]. Most of this work has focused on modeling small deformations of stiff structural materials.

More recently, offline animation and real-time simulation of deformable objects has entered the realm of computer graphics, as described in Gibson and Mirtich [6]. Mass-spring models have been common for simulating a diverse array of objects, including human tissues such as muscles [14] and blood vessels [3]. Mass-spring models are relatively easy to implement. However, they not only discretize the object into a set of finite point masses, they also discretize the equations of motion.

Unlike the mass-spring model, the finite element method is based on the equations of continuum mechanics. The feasibility and potential of this approach for computer animation was demonstrated by Terzopolous et. al. in 1987 [13]. Since, then, FEM has been applied to numerous graphical applications, including facial surgery [8] and fractures of rigid objects [10].

Real-time performance for surgery simulation of the human liver using FEM was achieved by Stéphane Cotin [4]. They modeled tissue as a linearly elastic material and allowed only small quasi-static deformations. In addition, their method requires a large preprocessing step, which is not appropriate for applications such as real-time planning of surgical procedures.

Another physically based method by James and Pai [7] that achieved real-time performance relies on the boundary element method (BEM), which reduces the dimensionality of the problem by considering only the surface of the object. However, this simplification may not be well suited for an application such as surgery simulation because it cannot properly handle non-homogenous solid materials or forces applied in the interior of the object.

Modeling large deformations requires quadratic strain, which generates a nonlinear system of differential equations. Recent work has achieved real-time performance for large deformations using reasonably sized meshes, including Yan Zhuang [16] and Picinbono et. al. [11]. Zhuang’s algorithm uses two key approximations: mass lumping and a graded mesh. These

techniques are effective to achieve real-time performance but may not be suitable for some applications because of a loss of realism.

Numerous publications in the FEM literature attempt to measure the a posteriori error, as surveyed in [17] and [1]. One such metric, the relative energy norm error, is used in this paper as described in section III. However, most of these error metrics are geared toward applications such as mesh refinement or shape function analysis rather than the visual or geometric accuracy of the deformation.

Unfortunately, each solving algorithm for simulating deformable objects has been demonstrated using its own set of test cases, making it difficult to compare the various approximation techniques. One example test case, the bending rod, has been used previously in the literature, including [16], [11], and [5]. However, as with other tests, this test has been implemented with different material properties and settings. This paper provides a consistent method to compare algorithms for simulating deformable objects.

III. ERROR METRICS

This paper describes three major classes of errors in the calculation of deformations: energy, surface, and interior error. The *relative energy norm* is commonly used in the FEM literature, but has no intuitive geometric meaning. The *surface error metric* is useful for applications such as animation where only the exterior shape of the object is visible. The *interior error metric* considers the error of both exterior and interior points of an object.

Let M_0 be the initial mesh of an object. For a particular time t , let M_t represent the correctly deformed mesh of the object based on physical laws. In this paper, we assume all material properties of the object are known, so M_t is unique. Let M_t' represent the deformed mesh calculated by an approximate numerical method. Let $\text{Int}(M_t)$ be a set containing all the points in the interior of meshed object M_t . Let $|S|$ represent the area spanned by the points in set S .

A. Relative Energy Norm Error Metric

The relative energy norm error metric is commonly used in the FEM literature [17]. Let \mathbf{u} be a vector containing the displacement of each node at time t . The energy norm for the error in the displacement is given by:

$$\|\mathbf{e}\| = \sqrt{(\mathbf{u}' - \mathbf{u})^T (\mathbf{u}' - \mathbf{u})}$$

The energy norm of the displacement is given by:

$$\|\mathbf{u}\| = \sqrt{\mathbf{u}^T \mathbf{u}}$$

Hence, the relative energy norm error E_n is then defined as:

$$E_n = \frac{\|\mathbf{e}\|}{\|\mathbf{u}\|} \quad (1)$$

This error is mathematically simple to calculate and manipulate, but is not easy to visualize geometrically.

B. Surface Error Metric

The surface error metric calculates the error of the boundary of a deformed object. Let the set operator \oplus represent the symmetric difference (also known as the xor) operator. At time t , the surface error of M_t' with respect to M_t is:

$$E_s = \frac{|\text{Int}(M) \oplus \text{Int}(M')|}{|\text{Int}(M)|}$$

Graphically, this error in 2D is visualized by superimposing the deformed objects and determining the symmetric difference of their areas, as shown in Figure 1. The error metric generalizes to 3D by using volumes instead of areas.

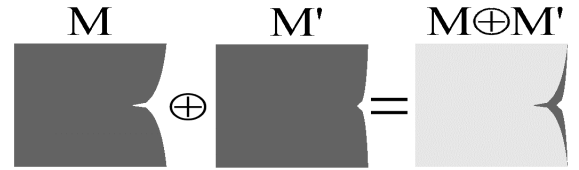


Figure 1: The dark area of object $M \oplus M'$ is the symmetric difference of objects M and M' .

C. Interior Error Metric

The interior error metric calculates the error of the displacements of all points that compose a deformed object. Let $\mathbf{p}_0 = (x, y)$ be a point in $\text{Int}(M_0)$. Let \mathbf{p} and \mathbf{p}' be the new location of \mathbf{p}_0 in M_t and M_t' , respectively, after a deformation. Let $d(\mathbf{a}, \mathbf{b})$ be a distance metric between points \mathbf{a} and \mathbf{b} .

$$E_i = \frac{1}{\sqrt[3]{|\text{Int}(M_0)|}} \int_{M_0} d(\mathbf{p}, \mathbf{p}') dM_0$$

The interior error is normalized by the square root of the area, or cube root of the volume, so that this normalization constant has dimension of length. Since the Euclidean distance metric is appropriate in this case, the error formula for 2D becomes:

$$E_i = \frac{1}{\sqrt[2]{|\text{Int}(M_0)|}} \iint_{(x,y) \in M_0} \sqrt{(x'-x)^2 + (y'-y)^2} dx dy$$

Unlike the relative energy norm, which sums the error of each DOF independently, the interior error metric sums the error of each point's displacement in 2D or 3D.

To implement this error metric efficiently, we convert the integral to a summation by selecting the location of the nodes of the mesh M_0 as a finite set of points that represent $\text{Int}(M_0)$. This is valid since values for the points

in the interior of the elements in the mesh are computed as linear interpolations of values at the element's nodes.

IV. BENCHMARKS

We propose two benchmarks: a 2D block being deformed by a needle and a 3D deformable rod fixed at one edge under the influence of gravity. These test cases can demonstrate how effectively the solving algorithms handle statics, dynamics, and deformations of different sizes.

A. 2D Block and Needle Benchmark

The first test simulates a 2D block of soft elastic material being poked from one side by a needle. Three sides of the square block are fixed. The displacement applied at a point causes a deformation, as shown in Figure 2. This test is relevant because it is the first step to simulating more complicated procedures, such as piercing soft tissue with a needle during surgery simulation.

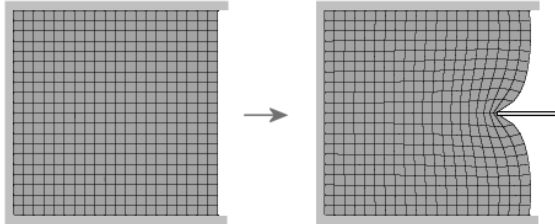


Figure 2: Block and needle benchmark. The top, bottom, and left sides are fixed. A needle pokes the block on the right.

The test is performed on two different materials defined by *MP1* in Table 1 and *MP2* in Table 2. *MP2* is based on real data for human prostate glandular tissue as specified in [9] and [12]. The damping constants are set to the minimum possible values that allow convergence for all the solving algorithms being tested, with exceptions covered in section VI.

| Material Property | Value |
|---------------------------|------------------------|
| Young's Modulus | 200 Pa |
| Poisson Ratio | 0.47 |
| Height | 1 m |
| Width | 1 m |
| Density | 2000 kg/m ³ |
| Rayleigh Damping α | 1.0 |
| Rayleigh Damping β | 0.1 |

Table 1: Material Properties #1 (*MP1*).

| Material Property | Value |
|---------------------------|------------------------|
| Young's Modulus | 60,000 Pa |
| Poisson Ratio | 0.495 |
| Height | 1 m |
| Width | 1 m |
| Density | 1140 kg/m ³ |
| Rayleigh Damping α | 1.0 |
| Rayleigh Damping β | 0.1 |

Table 2: Material Properties #2 (*MP2*). These properties are based on measured data for human prostate gland tissue [9, 12].

For the test, the needle is initially positioned on the right side of the block at the center of the non-fixed edge. At time 0, the needle begins moving at a speed of 0.08m/s to the left for 2 seconds, and then stops. A fixed time-step of $h=1/30$ seconds is used.

B. 3D Bending Rod Test Case

The second test involves simulating a 3D rod with a square cross-section. The rod is composed of a soft elastic material and is fixed at one end. The rod is subject to gravity in a viscous medium causing it to bend as shown in Figure 3.

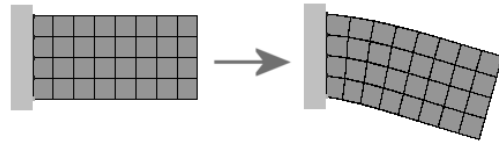


Figure 3: A 2D projection of the bending rod test case.

The material properties of the rod are described in Table 3. Initially, the rod is unaffected by gravity. At time 0, the force due to gravity is turned "on," with a constant of acceleration of 5 cm/s². A fixed time-step of $h=1/30$ seconds is used.

| Material Property | Value |
|---------------------------|-------------------------|
| Young's Modulus | 7000 Pa |
| Poisson Ratio | 0.2 |
| Height | 0.25 m |
| Width | 0.5 m |
| Depth | 0.25 m |
| Density | 10000 kg/m ³ |
| Rayleigh Damping α | 1.0 |
| Rayleigh Damping β | 0.1 |

Table 3: Material Properties #3 (*MP3*).

V. COMPUTING DEFORMATIONS

The algorithms tested in this paper use the finite element method (FEM) to compute soft tissue deformations. We assume the material being modeled is linearly elastic and only small deformations will result. The soft tissue is defined by a mesh composed of m elements created using n total nodes, each with e degrees of freedom, where $e=2$ for 2D or $e=3$ for 3D. The FEM problem is defined by a system of $d=en$ linear differential equations:

$$\mathbf{M} \mathbf{a}_i + \mathbf{C} \mathbf{v}_i + \mathbf{K} \mathbf{u}_i = \mathbf{f}_i \quad (1)$$

where \mathbf{M} is the mass matrix, \mathbf{C} is the damping matrix, \mathbf{K} is the stiffness matrix, \mathbf{f}_i is the external force vector, \mathbf{a}_i is the nodal acceleration vector, \mathbf{v}_i is the nodal velocity vector, and \mathbf{u}_i is the nodal displacement vector at time-step i [17].

The vector \mathbf{f}_i represents the forces exerted on the tissue. The matrices \mathbf{M} , \mathbf{C} , and \mathbf{K} are properties of the material being modeled and are constructed by superimposing the element mass, damping, and stiffness matrices [17]. The time integration techniques described below are used to solve for \mathbf{a}_i , \mathbf{v}_i , and \mathbf{u}_i for each time-step i .

A. Implicit Time Integration

To integrate the differential system (1) over time, we use the Newmark method [15], which translates the differential system into a linear system of equations. The method includes parameters β and γ that determine the properties of the resulting linear systems. Let h be the time-step duration. Displacement and velocity for the next time-step are approximated as:

$$\mathbf{u}_{i+1} = \mathbf{u}_i + h \mathbf{v}_i + (1-\beta) (h^2/2) \mathbf{a}_i + \beta (h^2/2) \mathbf{a}_{i+1}$$

$$\mathbf{v}_{i+1} = \mathbf{v}_i + (1-\gamma) h \mathbf{a}_i + \gamma h \mathbf{a}_{i+1}$$

When $2\beta \geq \gamma \geq 1/2$, the time integration is unconditionally stable. The error due to the numeric integration is $O(h^2)$ when $\gamma=1/2$, and $O(h)$ otherwise [15].

By setting the Newmark method parameters $\beta=0.5$ and $\gamma=0.5$, we obtain the implicit system:

$$(\mathbf{M} + h \mathbf{C}/2 + h^2 \mathbf{K}/4) \mathbf{a}_{i+1} = \mathbf{f}_{i+1} - (h \mathbf{C}/2 + h^2 \mathbf{K}/4) \mathbf{a}_i - (\mathbf{C} + h \mathbf{K}) \mathbf{v}_i - \mathbf{K} \mathbf{u}_i$$

$$\mathbf{v}_{i+1} = \mathbf{v}_i + (1/2) h (\mathbf{a}_i + \mathbf{a}_{i+1})$$

$$\mathbf{u}_{i+1} = \mathbf{u}_i + h \mathbf{v}_{i+1} + (1/4) h^2 (\mathbf{a}_i + \mathbf{a}_{i+1})$$

This is an implicit system since the next position is determined using the next velocity and acceleration. However, a linear system must be solved to obtain the acceleration vector.

B. Explicit Time Integration and Mass Lumping

Because of the linear system that must be solved at every time-step, the implicit integration formulation above can only achieve real-time interactive performance for small meshes. An alternative approach is to set the Newmark method parameters to $\beta=0$ and $\gamma=0.5$ to obtain the explicit system:

$$\mathbf{u}_{i+1} = \mathbf{u}_i + h \mathbf{v}_i + (1/2) h^2 \mathbf{a}_i$$

$$(\mathbf{M} + h \mathbf{C}/2) \mathbf{a}_{i+1} = \mathbf{f}_{i+1} - \mathbf{K} \mathbf{u}_{i+1} - \mathbf{C} (\mathbf{v}_i + h \mathbf{a}_i/2)$$

$$\mathbf{v}_{i+1} = \mathbf{v}_i + (1/2) h (\mathbf{a}_i + \mathbf{a}_{i+1})$$

This system is explicit since the next displacement vector depends only on current information. Unfortunately, this system requires solving a linear system for acceleration. However, we can use mass lumping, which approximates the continuous material as a particle system with mass lumped at the nodes, to diagonalize the \mathbf{M} and \mathbf{C} matrices. This decouples the system of equations into a set of algebraic equations that can quickly be solved [16, 11].

C. Quasi-Static Time-steps

With quasi-static time-steps, the deformation of the model at time time-step $i+1$ does not depend on any past history.

$$\mathbf{K} \mathbf{u}_{i+1} = \mathbf{f}_{i+1} \quad (2)$$

The quasi-static form of the problem finds the steady state solution \mathbf{u}_{i+1} , if it exists. The dynamic formulation should converge to this solution if the applied force is not modified, damping is present, and the system is stable.

This method implicitly assumes that acceleration and velocity are negligible between time-steps and visco-elastic material behavior cannot be modeled. However, very fast performance can be achieved using this quasi-static time-steps. But by taking advantage of the linearity of the system as well as the superposition principle, [4] achieved interactivity for very large meshes with this method.

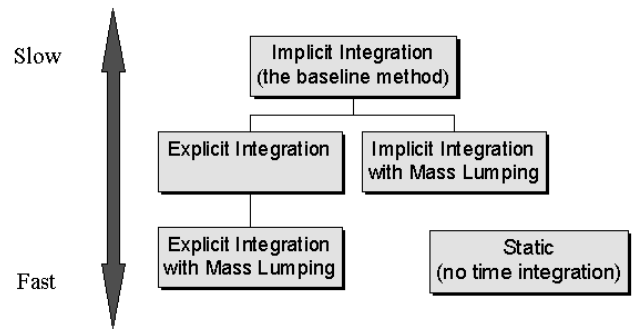


Figure 4: A summary of the relative speed and relationships between the FEM solving algorithms implemented and tested in this paper.

VI. NUMERICAL EXPERIMENT RESULTS

We implemented the FEM algorithms to compute deformations for linear elastic materials. The objects being simulated were defined using meshes composed of 2D quadrilateral or 3D hexahedral elements. We implemented the Gauss-Seidel iteration method to solve linear systems for the FEM solvers. The system was implemented in C++ and all tests were completed on a 750MHz PC with 128MB RAM.

The relative energy norm, surface, and interior error metrics were implemented to test the FEM solver methods described above. The correctly deformed mesh M was selected to be the deformation calculated by implicit integration without mass lumping. This solver algorithm does not rely on any major approximations so it is most representative of the real-world outcome. Since the goal here is not to verify the correctness of the FEM formulation but rather the relative merits of computational speed-up algorithms, this is deemed an appropriate choice.

A. Direct Comparison of Solving Algorithms

The block and needle test results for *MPI* are shown for the relative energy norm error in Figure 5, the surface error in Figure 6, and the interior error in Figure 7 and Figure 8. All graphs assume that implicit integration determines the correct deformation. In all cases, the same general trend holds. Explicit integration with no lumping is the best approximation followed by the mass lumping methods. Quasi-static consistently performs worst.

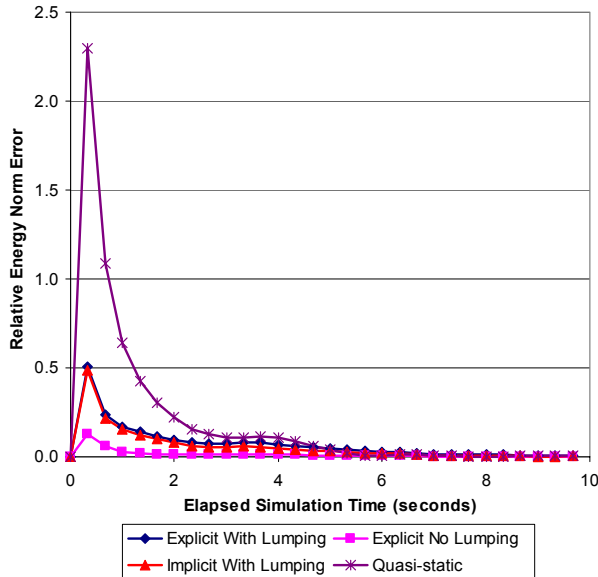


Figure 5: Relative energy norm error for the block and needle test with *MPI*.

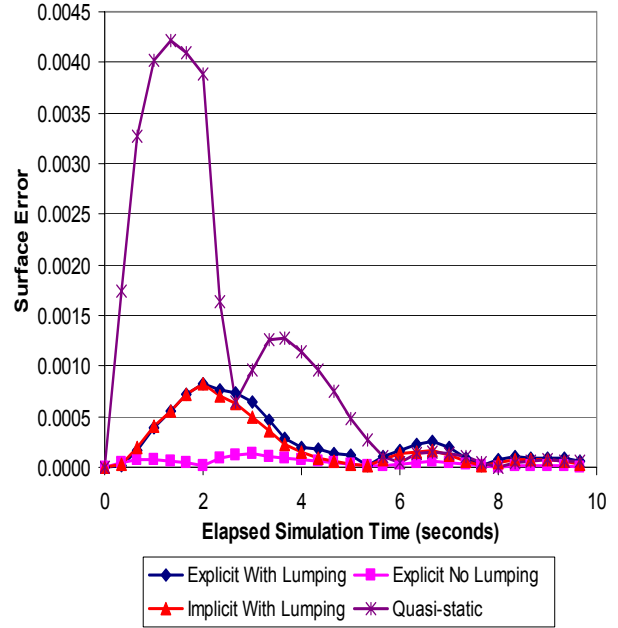


Figure 6: Surface error for the block and needle benchmark with *MPI*.

The multiple peaks in Figure 6 after 5 seconds represent the oscillations of the soft elastic material as it settles into its steady state configuration. These oscillations occur because the system is not perfectly damped.

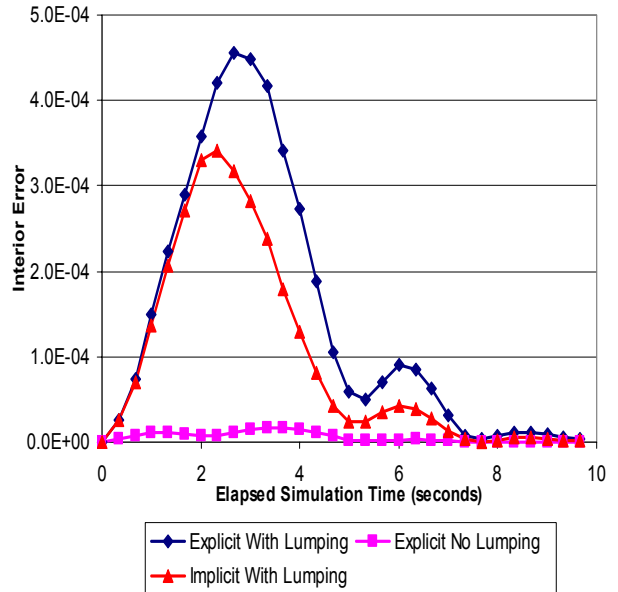


Figure 7: Interior error for the block and needle benchmark with *MPI*.

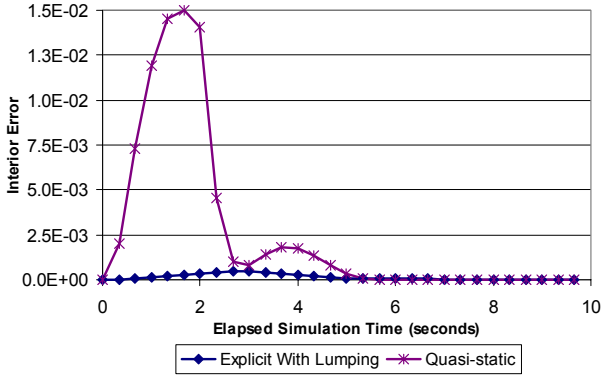


Figure 8: Interior error for the block and needle benchmark with *MPI*. Quasi-static performs significantly worse than explicit with mass lumping.

Quasi-static time-steps performed worst in all cases, with the error being off the chart for most of the simulations. This error is high because the material being modeled is visco-elastic, so the assumptions of quasi-static time-steps from section **Error! Bookmark not defined.** are not satisfied.

Explicit integration with mass lumping performed next worst. This is most likely because the equations of continuum mechanics are modified. Also, the time integration scheme decoupled the system of equations, causing information to travel slowly through the object. The artificially slow rate of information transfer causes the dynamics of this simulation to be inaccurate. Also interesting to note is that the explicit integration with mass lumping error is roughly the sum of the error due only to mass lumping or only to explicit integration.

The experiment confirms that all the time integration schemes converge to the same final solution, which is the static solution that is independent of damping or mass distribution.

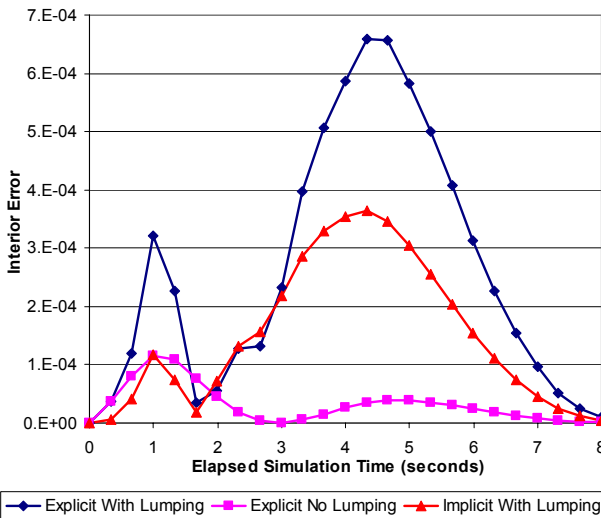


Figure 9: Interior error for the bending rod benchmark.

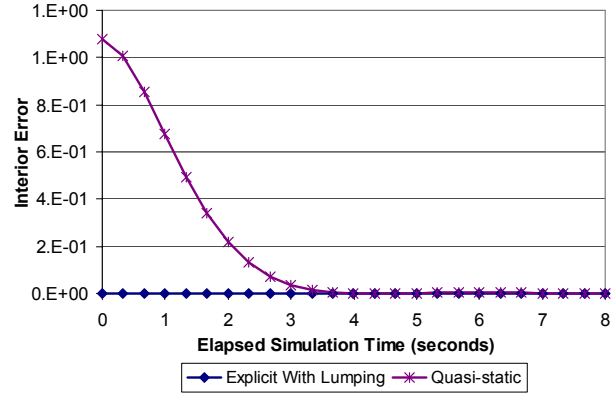


Figure 10: Interior error for the bending rod benchmark.

Figure 9 and Figure 11 show the bending rod benchmark results for the interior error metric. The results for this test follow the same general trends as with the block poking test case. Since quasi-static reaches its steady-state configuration after the first time-step, its error steadily decreases as time passes and the solutions of the transient integration models approach steady state.

In the bending rod test, the elastic rod bends and overshoots the steady-state configuration before settling into the final configuration. Since the rod is not perfectly damped, it oscillates around the steady-state configuration. In particular, at 4.2 seconds, the rod drops below the steady-state configuration, which explains why the error of the quasi-static algorithm approaches 0 at that time.

B. Solving Algorithms and Material Properties

The material properties have a significant impact on the solving algorithms. *MPI* was set so that all the solving algorithms covered in this paper would converge. However, increasing the stiffness or incompressibility of *MPI* causes the explicit time integration methods to fail to converge. The systems will perpetually gain energy until the node displacements diverge to infinity.

This reveals a well-known problem: explicit integration is unstable for large time-steps when simulating stiff materials [2]. The maximum stable time-step length is inversely proportional to the natural frequency of the dynamic system (1), which increases for stiffer materials. Therefore, using explicit integration with *MP2* is impossible for a time-step of $h=1/30$ seconds. For stable convergence with any reasonable damping, the time-step h must be reduced to less than 10^{-6} seconds, which makes real-time performance impossible on any PC available today. On the other hand, the implicit integration method converged well with damping constants α and β both under 1, although several thousand Gauss-Seidel iterations were required to accurately solve the linear system for $h=1/30$ seconds.

Empirically, numerical instabilities arise for large time-

steps when the Poisson ratio approaches its maximum value of 0.5, which represents an incompressible material. To counteract this problem, Picinbono et. al. [11] propose using a lower Poisson ratio but adding a penalty force. The artificial force at each node is directed normal to the opposite face of the element and serves to penalize element compression. No formal methods are given to justify the realism of this artificial force, or how to relate this force to the Poisson ratio for a given material.

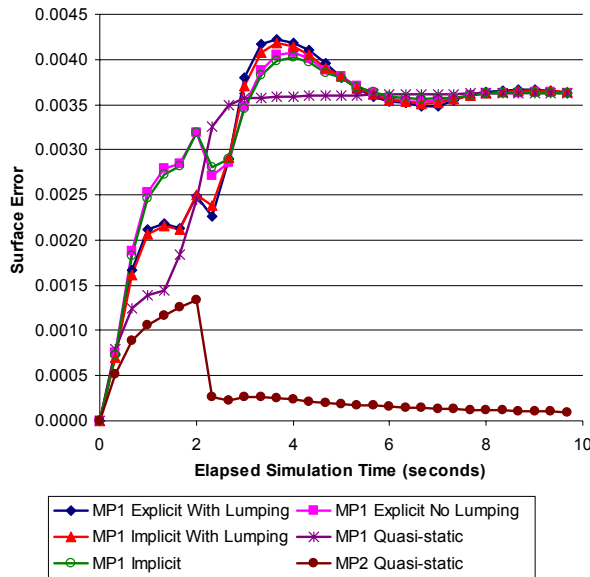


Figure 11: Surface error for the block and needle test assuming implicit integration with real prostate tissue data (*MP2*) is used as the correct deformation. Quasi-static is the only approximation using *MP2* that numerically converged. The results using *MP1* are also compared, but they converge to a different solution.

Figure 11 uses *MP2* with implicit integration as the correct deformation. It compares the results with the quasi-static algorithm using *MP2* as well as all the solving methods using *MP1*. The methods using *MP1* converge to a different steady-state solution, which is intuitive since materials with different compressibilities will react differently to identical forces or displacement constraints. This explains the error at the end of the simulation. Also, the quasi-static algorithm with *MP2* has far less error than the algorithms using *MP1*. This shows that the accuracy of the material properties is of greater importance than the choice of solver algorithm.

In conclusion, material properties impact the effectiveness of an algorithm. In particular, approximations such as explicit integration and mass lumping do not work well for stiff and incompressible materials.

VII. FUTURE WORK

In the future, we hope to compare the various FEM algorithms to physical experimental results rather than to

numerical approximations. We plan to test quadratic strain FEM models and generate special test cases that highlight the difference between linear and nonlinear models when large deformations are created. We also hope to develop an error metric that tests how sensitive an algorithm is to changes in the FEM formulation parameters, such as material properties, mesh density, and mesh shapes.

To simulate incompressible materials in real time, Picinbono et. al. [11] propose using penalty forces. We would like to test the accuracy of this approach with a large time-step by comparing it to using a high Poisson ratio with a small time-step.

The mass-lumping approach also requires that the damping matrix be a scale factor times the lumped mass matrix, rather than using the traditional Rayleigh damping form. This implies that there are restrictions on the nature of damping that is supported by this method. It would be useful to understand the physical side effects of restricting the damping matrix to a diagonal matrix.

VIII. ACKNOWLEDGEMENTS

We thank Jean Pouliot for providing medical background and K. Gopalakrishnan for his valuable feedback.

IX. REFERENCES

- [1] Mark Ainsworth and J. Tinsley Oden. A posteriori error estimation in finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 142 (1997) 1-88.
- [2] David Baraff and Andrew Witkin. Large Steps in Cloth Animation. SIGGRAPH, 1998.
- [3] Joel Brown, Stephen Sorkin, Cynthia Bruyns, Jean-Claude Latombe, Kevin Montgomery, Michael Stephanides. Real-Time Simulation of Deformable Objects: Tools and Applications. *Computer Animation*, Seoul, Korea, November 2001.
- [4] Stéphane Cotin, Hervé Delingette, and Nicholas Ayache. Real-Time Elastic Deformations of Soft Tissues for Surgery Simulation. *IEEE Transactions on Visualization and Computer Graphics*. Vol. 5, No. 1, January-March 1999.
- [5] Gilles Debunne, Mathieu Desbrun, Alan Barr, and Marie-Paule Cani. Interactive Multiresolution Animation of Deformable Models. 1999.
- [6] Sarah F. F. Gibson and Brian Mirtich. A Survey of Deformable Modeling in Computer Graphics. MERL, TR-97-19, November 1997.
- [7] Doug L. James and Dinesh K. Pai. ArtDefo: Accurate Real Time Deformable Objects. In ACM Computer Graphics, August 1999. ACM SIGGRAPH, 1999.
- [8] R. M. Koch, M.H. Gross, F.R. Carls, D.F. von Büren, G. Frankhauser, Y. I. H. Parish. Simulating Facial Surgery Using Finite Element Models. 1996.
- [9] Thomas A. Krouskop, Thomas M. Wheeler, Faouza Kallel, Brian S. Garria, and Timothy Hall. Elastic Moduli of Breast and Prostate Tissues Under Compression. *Ultrasonic Imaging*, Volume 20, pages 260-274. 1998.
- [10] James F. O'Brien and Jessica K. Hodgins. Graphical Modeling and Animation of Brittle Fracture. 2000.
- [11] G. Picinbono, H. Delingette, N. Ayache, Nonlinear and anisotropic elastic soft tissue models for medical simulation.

IEEE International Conference on Robotics and Automation,
Seoul, South Korea, May 21-26, 2001.

- [12] Dee-Ann Radford, David S. Followill and William F. Hanson. Design of an Anthropomorphic Intensity Modulated Radiation Therapy Quality Assurance Phantom. *AAPM*, 2001.
- [13] Demetri Terzopolous, John Platt, Alan Barr, Kurt Fleischer. Eastically Deformable Models. *Computer Graphics*, Volume 21, Number 4. July 1987.
- [14] K. Waters. A Muscle Model for Animating Three-Dimensional Facial Expression. In *Computer Graphics Proceedings, Annual Conference Series, Proceeding of SIGGRAPH 87*, pages 17-24. ACM SIGGRAPH 1987.
- [15] W. L. Wood. Some Transient and Coupled Problems – A State-of-the-Art Review. Chapter 8, *Numerical Methods in Transient and Coupled Problems*, pp 149-177. R. W. Lewis, E. Hinton, P. Bettess, and B. A. Schrefler, editors. 1987.
- [16] Yan Zhuang. Real-time Simulation of Physically Realistic Global Deformations. PhD. Thesis. 2000.
- [17] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method*. Fifth Edition. Butterworth-Heinemann. 2000.