Interactive Simulation of Surgical Needle Insertion and Steering

Nuttapong Chentanez
U.C. Berkeley

Ron Alterovitz
U.N.C. Chapel Hill

Daniel Ritchie
U.C. Berkeley

Lita Cho
U.C. Berkeley

Kris K. Hauser
U.C. Berkeley

Ken Goldberg
U.C. Berkeley

Jonathan R. Shewchuk
U.C. Berkeley

James F. O'Brien
U.C. Berkeley

Abstract

We present algorithms for simulating and visualizing the insertion and steering of needles through deformable tissues for surgical training and planning. Needle insertion is an essential component of many clinical procedures such as biopsies, injections, neurosurgery, and brachytherapy cancer treatment. The success of these procedures depends on accurate guidance of the needle tip to a clinical target while avoiding vital tissues. Needle insertion deforms body tissues, making accurate placement difficult. Our interactive needle insertion simulator models the coupling between a steerable needle and deformable tissue. We introduce (1) a novel algorithm for local remeshing that quickly enforces the conformity of a tetrahedral mesh to a curvilinear needle path, enabling accurate computation of contact forces, (2) an efficient method for coupling a 3D finite element simulation with a 1D inextensible rod with stick-slip friction, and (3) optimizations that reduce the computation time for physically based simulations. We can realistically and interactively simulate needle insertion into a prostate mesh of 13,375 tetrahedra and 2,763 vertices at a 25 Hz frame rate on an 8-core 3.0 GHz Intel Xeon PC. The simulation models prostate brachytherapy with needles of varying stiffness, steering needles around obstacles, and supports motion planning for robotic needle insertion. We evaluate the accuracy of the simulation by comparing against real-world experiments in which flexible, steerable needles were inserted into gel tissue phantoms.

Keywords: surgical simulation, needle insertion, real-time finite element methods, coupled simulation

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.6.8 [Simulation and Modeling]: Types of Simulation—Animation.

1 Introduction

Needle insertion is an essential component of many clinical procedures such as biopsies, injections, neurosurgery, and brachytherapy cancer treatment [Abolhassani et al., 2007]. The success of these procedures depends on how close the needle tip is maneuvered to the target. It is crucial that the needle avoid bone and other critical structures and organs [Kohn et al., 2000]. Unfortunately, needle insertion deforms body tissues enough that poor accuracy is the norm in practice. For example, experienced physicians inserting radioactive seeds into the prostate gland for brachytherapy prostate cancer treatment experience average placement errors of 6.3 mm, about 15% of the prostate's diameter [Taschereau et al., 2000].

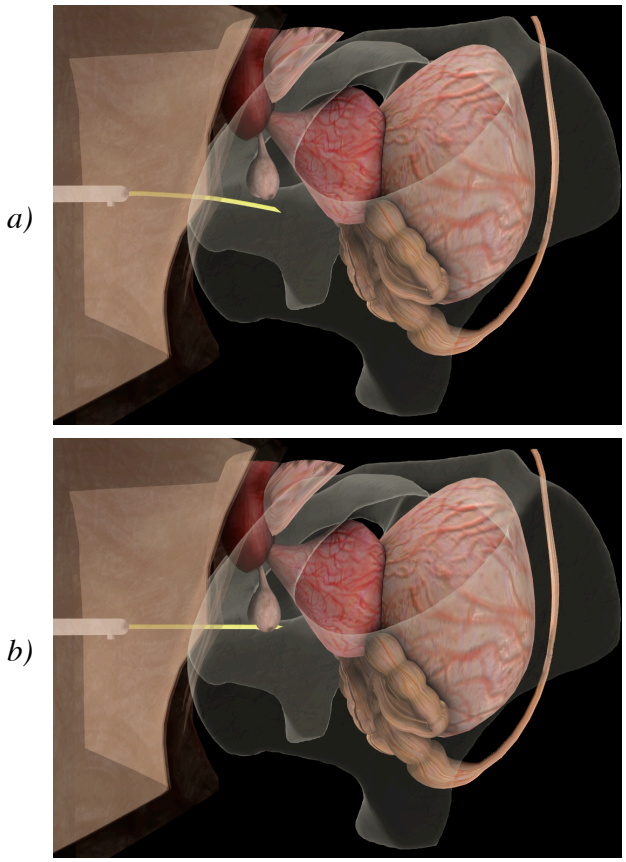

Figure 1: *Screenshots from our prostate brachytherapy simulator. A needle is inserted from the left through the epidermis and dermis into the prostate gland. a) Bevel-tip flexible needle. b) Symmetric-tip stiff needle.*

Computer simulations of needle insertion procedures enable physicians and other clinicians to train in a controlled environment that exposes them to both common and rare patient cases without risks to patient safety. Studies indicate that surgical skills learned using computational simulators directly improve operating room performance by significantly decreasing procedure time and reducing the frequency of medical errors by up to sixfold compared to traditional training [Seymour et al., 2002; Satava, 2005; Gallagher et al., 2005]. Surgical simulations also have uses for pre-operative planning [Alterovitz and Goldberg, 2008; Taylor, 2006].

We present a new simulator that models tissue deformation, needle elasticity, and their interaction. It allows us to realistically simulate the deflections that occur as thin needles travel through inhomogeneous tissues. A motivation for modeling needle elasticity is a new class of flexible, steerable needles recently developed in collaboration between researchers at U.C. Berkeley and Johns Hop-

From the ACM SIGGRAPH 2009 conference proceedings.

kins University [Webster III et al., 2005b; Webster III et al., 2006]. These bevel-tip steerable needles have a flexible shaft that curves as it penetrates soft tissue, due to asymmetric forces exerted at the needle's bevel tip. By twisting the needle as it is inserted, a physician can steer its tip around obstacles to reach clinical targets in soft tissues [Alterovitz et al., 2005; Alterovitz et al., 2007]. It is not easy to learn how to control steerable needles, and realistic training simulations will accelerate their deployment in clinical practice.

Several impediments make it difficult to simulate the interaction between a needle and soft tissues: a static spatial discretization (*e.g.* a fixed finite element mesh) does not easily support the accurate computation of contact forces and needle steering; the mismatch between needle stiffness and tissue stiffness hinders numerical stability; and the simulation must run at interactive rates. To address these challenges, we introduce (1) a novel algorithm for local remeshing, (2) an efficient algorithm for coupling a 3D finite element simulation and a 1D elastic rod simulation with stick-slip friction, and (3) several generally applicable optimizations for reducing computation time for physically based simulations. Our remeshing algorithm efficiently relocates and creates nodes so they lie along a curvilinear needle path in a volumetric mesh, enabling the simulation to apply cutting and frictional forces along the needle shaft at mesh nodes while maintaining a high quality tetrahedral mesh for computing tissue deformations. Our optimizations include accelerating the solution of the linear complementarity problem for node friction states, using a parallel conjugate gradient method on sparse matrices, and using a parallel lazy update of Jacobian matrices for tetrahedral elements.

Together, these algorithms and enhancements enable us to realistically simulate needle insertion in the prostate at interactive frame rates. We achieve frame rates of 25 Hz on an 8-core 3.0 GHz Intel Xeon PC for a prostate mesh of 13,375 tetrahedra and 2,763 vertices. We use realistic material properties for human tissue, making it more challenging than the more compliant materials for which real-time performance is usually reported. We present simulations of prostate brachytherapy with needles of different stiffness, simulations of needles steered around obstacles, and an application to motion planning for robotic needle insertion. Throughout our trials, our remeshing procedure consistently maintained high mesh quality. We evaluate the accuracy of the simulator using data extracted from real-world experiments in which flexible, steerable needles were inserted into gel tissue phantoms. The simulated and real-world deformations are qualitatively and quantitatively similar.

2 Background

Surgical simulators often use methods originating in computer graphics for fast simulation of deformable tissues. Deformable body simulation has been a topic of active research in the computer graphics community since Terzopoulos et al. [1987]. The finite element method has been extended to handle fracture [O'Brien and Hodgins, 1999], viscoplastic behavior [O'Brien et al., 2002; Goktekin et al., 2004; Bargteil et al., 2007; Wojtan and Turk, 2008], and inverted elements [Irving et al., 2004; Irving et al., 2007]. Researchers have attained real-time performance with multiresolution methods [Debunne et al., 2000] and stiffness warping techniques [Müller et al., 2002; Müller and Gross, 2004].

Several open-source surgical simulation toolkits model tissue deformation, including GiPSi [Çavuşoğlu et al., 2006] and SOFA [Allard et al., 2007]. Surgery simulators must also model tool-tissue interactions. Finite element methods have been developed that use remeshing [Nienhuys and van der Stappen, 2001; Mendoza and Laugier, 2003; Picinbono et al., 2003] or discontinuous basis functions [Lindblad and Turkiyyah, 2007] to simulate tissue cutting.

Abolhassani et al. [2007] survey needle insertion modeling and simulation. Physically based simulations have been developed

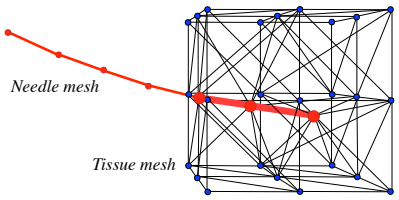

Figure 2: *The needle mesh (red) and tissue mesh (black). The large red nodes and bold red edges belong to both meshes and couple the two objects.*

for the insertion of rigid needles [Alterovitz et al., 2003], flexible symmetric-tip needles [DiMaio and Salcudean, 2005], and flexible bevel-tip needles [Alterovitz et al., 2005] into 2D slices of tissue. Simulations of 3D rigid needle insertion have been developed using mass-spring models [Marchal et al., 2006; Vidal et al., 2008] and the non-physically based chain-mail model [Wang and Fenster, 2004]. We know of no prior simulation that handles interactive simulation of coupled tissue and needle deformation in 3D.

For 3D finite element simulation of needle insertion, mesh maintenance is a key difficulty. Nienhuys and van der Stappen [2004] use a finite element method and recursive refinement of the mesh until mesh nodes are very close to the needle shaft. Goksel et al. [2006] simulate interactive 3D rigid needle insertion in linear elastic tissues using local remeshing with node snapping and face splitting. Dehghan and Salcudean [2007] extend this method to support nonlinear material properties. Our new remeshing algorithm builds on this body of work by using additional remeshing operations (edge and tetrahedron splitting) and by choosing operations based on mesh quality, not on geometric heuristics.

Alternatively, different meshes can be tied together by binding constraints without remeshing, as Sifakis et al. [2007] do. However, in real tissues, the deformation gradient around the needle shaft is discontinuous, with a near-singularity at the needle tip. Therefore, it is essential for accuracy that our volume mesh conform to the needle, and especially that it have a node at the needle tip. Nonconforming approaches distribute the needle forces onto nearby nodes, so the largest deformation does not coincide with the needle and the piecewise constant strains are inaccurate for elements that intersect the needle. While the bulk tissue behavior would be similar away from the needle, the needle path would be quite different.

We model the needle as a 1D rod bending in 3D space. Pai [2002] uses Cosserat theory to simulate an elastic rod as a boundary value problem. Bertails et al. [2006] improved the running time of this approach to $O(n^2)$ per time step and later to $O(n)$ [Bertails, 2009], where n is the number of vertices used to discretize the needle. The static equilibrium of the rod is considered by Loock and Schömer [2001] and Grégoire and Schömer [2006]. Spillmann and Teschner [2007] augment the method to include dynamics and be solved in $O(n)$ time per time step. Our simulation builds on the method of Bergou et al. [2008], who simplify the computation by assuming that the twist wave propagates at infinite speed through the rod.

3 Methods

We model tissue elasticity with constitutive equations discretized over a tetrahedral mesh by a finite element method. The needle has a small diameter, so we model it as a 1D elastic rod, following DiMaio and Salcudean [2005]. We denote the tetrahedral tissue mesh by T. The needle is represented by a mesh $\widetilde{T}$ comprising a subset of the edges of T, plus additional edges that represent the portion of the needle outside the tissue, as Figure 2 shows. We dynamically update the tetrahedral mesh so that it always conforms to the needle, as described in Section 3.4.

We use a stick-slip model of the friction between the tissue and the needle shaft. Each node shared by the two meshes is in either a

static or dynamic friction state. Static friction implies that the needle and tissue are moving in lockstep at the node; dynamic friction implies that they are sliding against each other. The needle tip is a special case, because when it is moving into the tissue with dynamic friction it is cutting tissue, and therefore encounters much greater resistance than accounted for by dynamic friction alone.

Our main contributions are the remeshing and coupling algorithms, which are independent of how the tissue and needle forces are computed. Thus, we summarize our choices of force computations only briefly, and refer the reader to the original papers for details. Algorithm 1 summarizes the execution of one simulation step.

Algorithm 1 Needle simulation (one time step)

1: Compute tissue forces $\hat{\mathbf{F}}$, needle forces $\tilde{\mathbf{F}}$, and Jacobians $\partial\mathbf{F}/\partial\mathbf{x}$ and $\partial\mathbf{F}/\partial\mathbf{v}$ for both tissue and needle (§3.2)
2: Solve coupled system Eq. (7), yielding the friction configurations and the nodal accelerations in tissue and needle (§3.3)
3: Update the positions and velocities of tissue and needle (§3.1)
4: **if** the needle tip is cutting or retracting **then**
5: Remesh around the tip (§3.4.1)
6: Reparameterize the needle (§3.4.2)

Throughout the simulation, we maintain for each node of T both a material coordinate (recording the geometry of the undeformed mesh) and a world coordinate (recording the deformed mesh). Let $\mathbf{u}_i^k$, $\mathbf{x}_i^k$, $\mathbf{v}_i^k$, $\mathbf{a}_i^k \in \mathbb{R}^3$ denote the material position, world position, velocity, and acceleration of the i^{th} node at time index k. We omit the node index to refer to a vector of properties for all the nodes. We omit the time index to refer to the current time. We use carets (ˆ) to denote tissue properties, and tildes (˜) for needle properties.

3.1 Implicit Time Integration

Let n be the number of nodes in the (tissue or needle) mesh. We integrate the node positions $\mathbf{x} \in \mathbb{R}^{3n}$ and velocities $\mathbf{v} \in \mathbb{R}^{3n}$ over time with Newmark's method,

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \triangle t\mathbf{v}^k + \triangle t^2\left(\left(\frac{1}{2}-\beta\right)\mathbf{a}^k + \beta\mathbf{a}^{k+1}\right), \quad (1)$$

$$\mathbf{v}^{k+1} = \mathbf{v}^k + \triangle t\left((1-\gamma)\,\mathbf{a}^k + \gamma\mathbf{a}^{k+1}\right), \quad (2)$$

where $\triangle t$ is the time step, $0 \le \beta \le 0.5$, and $0 \le \gamma \le 1$. (All our simulation results use $\beta = 0.25$ and $\gamma = 0.5$, equivalent to integration by the trapezoid rule.) We obtain the accelerations $\mathbf{a}^{k+1} \in \mathbb{R}^{3n}$ by solving

$$\mathbf{F}(\mathbf{x}^{k+1}, \mathbf{v}^{k+1}) = \mathbf{M}\mathbf{a}^{k+1}, \quad (3)$$

where $\mathbf{M} \in \mathbb{R}^{3n\times 3n}$ is the mass matrix and $\mathbf{F}(\cdot) \in \mathbb{R}^{3n}$ is the sum of all internal forces such as stiffness and damping forces (discussed in the next section) and external forces such as gravity. Because Equation (3) is nonlinear, we linearize it with one Newton–Raphson iteration; *i.e.* by solving

$$\mathbf{F}(\mathbf{x}^k, \mathbf{v}^k) + \frac{\partial\mathbf{F}}{\partial\mathbf{x}}(\mathbf{x}^{k+1} - \mathbf{x}^k) + \frac{\partial\mathbf{F}}{\partial\mathbf{v}}(\mathbf{v}^{k+1} - \mathbf{v}^k) \approx \mathbf{M}\mathbf{a}^{k+1}, \quad (4)$$

where $\partial\mathbf{F}/\partial\mathbf{x}$, $\partial\mathbf{F}/\partial\mathbf{v} \in \mathbb{R}^{3n\times 3n}$ are the Jacobian matrices of force with respect to position and velocity, evaluated at $(\mathbf{x}^k, \mathbf{v}^k)$.

Ignoring for now the coupling between needle and tissue, we substitute (1) and (2) into (4) to obtain sparse linear systems

$$\hat{\mathbf{A}}\hat{\mathbf{a}}^{k+1} = \hat{\mathbf{b}}, \quad (5)$$

$$\tilde{\mathbf{A}}\tilde{\mathbf{a}}^{*k+1} = \tilde{\mathbf{b}} \quad (6)$$

for the tissue and needle nodes' accelerations, respectively. The asterisk indicates that $\tilde{\mathbf{a}}^{*k+1}$ is a temporary quantity, for reasons explained in Section 3.4.2. Having solved for $\hat{\mathbf{a}}^{k+1}$ and $\tilde{\mathbf{a}}^{*k+1}$, we obtain $\hat{\mathbf{x}}^{k+1}$, $\hat{\mathbf{v}}^{k+1}$, $\tilde{\mathbf{x}}^{*k+1}$, and $\tilde{\mathbf{v}}^{*k+1}$ from Equations (1) and (2).

The sparsity of $\hat{\mathbf{A}}$ is unstructured, whereas $\tilde{\mathbf{A}}$ has bandwidth 5 (measured in 3×3 blocks) because each needle node has nonzero entries for the two nodes before and after it on the needle. We assemble both matrices and store $\hat{\mathbf{A}}$ with a block compressed sparse row format and $\tilde{\mathbf{A}}$ with a block banded format.

3.2 Force and Jacobian Computations

We follow Irving et al. [2004] in computing tissue forces. We compute a deformation gradient $\partial\hat{\mathbf{x}}/\partial\hat{\mathbf{u}}$ for each tetrahedron. Then we perform a singular value decomposition (SVD) of each element's deformation gradient and treat negative eigenvalues specially so that inverted tetrahedra, if they arose, would not be fatal to the simulation. We calculate the first Piola–Kirchhoff stress from the tetrahedron's deformation gradient and material properties, then compute the elastic force from the stress. We apply a damping force, computed similarly from the velocity gradient. To evaluate the Jacobians of the tissue forces, $\hat{\mathbf{F}}$, we differentiate the forces with respect to $\hat{\mathbf{x}}$ and $\hat{\mathbf{v}}$ while holding the SVD rotation matrices constant.

We borrow from two recent works to compute the needle forces, $\tilde{\mathbf{F}}$. We compute bending and twisting forces following Bergou et al. [2008]. The needle is inextensible, but instead of enforcing inextensibility by projection, we use a method of Spillmann and Teschner [2007] that applies compensatory stretching forces. In contrast to these two works, we treat the bending and stretching forces implicitly, which requires Jacobian computation. However, the twisting force has an asymmetric Jacobian. To avoid solving an asymmetric linear system, we integrate the twisting force explicitly by setting its contribution to the Jacobians to zero.

3.3 Needle-Tissue Coupling and Cutting

Each node i of $\widetilde{T}$ has a friction state s_i which is one of FREE, STATIC, DYNAMIC−, or DYNAMIC+. The sign of dynamic friction indicates the direction in which the needle is sliding along the tissue at that node. The FREE nodes are not in the tissue. The other nodes, which are shared by the tissue mesh T and the needle mesh $\widetilde{T}$, are called *coupling nodes*. Assume that each coupling node has the same index in both meshes.

If we know every state s_i, we can solve the coupled equations with Lagrange multipliers, introducing for each coupling node i an additional variable $\mathbf{c}_i \in \mathbb{R}^3$, the constraint force required to satisfy the stick-slip constraint. This force acts upon tissue and needle nodes i in equal magnitude but opposite directions. To accommodate dynamic friction and sliding of the needle, we express $\mathbf{c}_i$ in a local coordinate system of the needle, in which the first axis is the unit vector $\mathbf{t}_i$ that is tangential to the needle at node i. We approximate the tangent as $\mathbf{t}_i = \text{NORMALIZE}(\text{NORMALIZE}(\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_{i-1}) + \text{NORMALIZE}(\tilde{\mathbf{x}}_{i+1} - \tilde{\mathbf{x}}_i))$. Let $\mathbf{R}_i \in \mathbb{R}^{3\times 3}$ be the rotation matrix that transforms from local coordinates to world coordinates, so that $\mathbf{R}_i\mathbf{c}_i$ is the constraint force at the coupling node i in world coordinates. For a DYNAMIC node, we set the first column of $\mathbf{R}_i$ to zero, thus ignoring the tangential constraint force in direction $\mathbf{t}_i$.

The coupled system is

$$\begin{bmatrix} \hat{\mathbf{A}} & 0 & \hat{\mathbf{W}}\mathbf{R} \\ 0 & \tilde{\mathbf{A}} & -\tilde{\mathbf{W}}\mathbf{R} \\ (\hat{\mathbf{W}}\mathbf{R})^\mathsf{T} & -(\tilde{\mathbf{W}}\mathbf{R})^\mathsf{T} & \mathbf{Z} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{a}}^{k+1} \\ \tilde{\mathbf{a}}^{*k+1} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{b}} + \hat{\mathbf{W}}\mathbf{R}\mathbf{d} \\ \tilde{\mathbf{b}} - \tilde{\mathbf{W}}\mathbf{R}\mathbf{d} \\ \mathbf{e} \end{bmatrix}. \quad (7)$$

The first two rows are Equations (5) and (6) augmented with the constraint forces. Here, $\mathbf{d}_i$ is $[0,0,0]^\mathsf{T}$ for a STATIC node and $[s_i f_i, 0, 0]^\mathsf{T}$ for a DYNAMIC node, s_i is 1 for DYNAMIC+ or −1 for DYNAMIC−, f_i is the magnitude of dynamic friction (possibly including a cutting force f_{cut} at the needle tip), and $\hat{\mathbf{W}}$ and $\tilde{\mathbf{W}}$ are 0–1 matrices that map coupling nodes to the tissue nodes and needle

nodes, respectively. Thus, for a dynamic node i, the unknown tangential component of the constraint force $\mathbf{c}_i$ on the left-hand side is supplanted by the known friction $\pm f_i$ on the right.

The third row of Equation (7) constrains the coupling nodes to have the same positions in the tissue and needle meshes, except that a node in a dynamic friction state permits the needle to slide tangentially relative to the tissue. If s_i is STATIC, we constrain $\hat{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_i$ to be identical. If it is DYNAMIC, we constrain them to agree in the directions orthogonal to $\mathbf{t}_i$. Thus, $\mathbf{Z}$ is a diagonal matrix in which a diagonal entry is 1 for the tangential component of a DYNAMIC node and 0 otherwise, and $\mathbf{e}$ is the right-hand side of the equation found by substituting Equation (1) into

$$(\hat{\mathbf{W}}\mathbf{R})^{\mathsf{T}}(\hat{\mathbf{x}}^{k+1} - \hat{\mathbf{x}}^{k}) - (\tilde{\mathbf{W}}\mathbf{R})^{\mathsf{T}}(\tilde{\mathbf{x}}^{*k+1} - \tilde{\mathbf{x}}^{*k}) = \mathbf{0} \tag{8}$$

and moving the terms that include $\hat{\mathbf{a}}^{k+1}$ and $\tilde{\mathbf{a}}^{*k+1}$ to the left-hand side. Note that the columns having a 1 on $\mathbf{Z}$'s diagonal are the same columns of $\mathbf{R}$ that we set to zero.

The coupled system (7) is symmetric but indefinite. Indefinite systems usually cannot be solved by the conjugate gradient method (CG), and require algorithms like MINRES or SYMMLQ [Paige and Saunders, 1975], which are about twice as expensive per iteration. Nevertheless, we find that CG effectively solves our system (7) to the desired tolerance in practice. Marcia [2008] makes the same observation for linear systems similar to ours, and provides a partial explanation.

An advantage of our formulation (7) is that we can update it quickly if the friction states s_i change. The tricky part of stick-slip friction is that the states s_i are not known in advance (except the FREE ones). We must guess them, then guess again if we are wrong. We have guessed right if they satisfy the following constraints. For a STATIC node i except the needle tip,

$$-f_i \le [1\ 0\ 0]^{\mathsf{T}} \cdot \mathbf{c}_i \le f_i, \tag{9}$$

where f_i is the static friction threshold, which experimentally is the same as the dynamic friction magnitude. For a STATIC needle tip i,

$$-(f_i + f_{\text{cut}}) \le [1\ 0\ 0]^{\mathsf{T}} \cdot \mathbf{c}_i \le f_i. \tag{10}$$

For a DYNAMIC node i (needle tip or not),

$$s_i \mathbf{t}_i \cdot \left((\hat{\mathbf{x}}_i^{k+1} - \hat{\mathbf{x}}_i^{k}) - (\tilde{\mathbf{x}}_i^{*k+1} - \tilde{\mathbf{x}}_i^{*k}) \right) \ge 0; \tag{11}$$

that is, the relative tangential movement between tissue and needle DYNAMIC coupling nodes must not change direction.

The equations and constraints together form a linear complementarity problem (LCP). Given q coupling nodes, there are 3^q possible settings of the friction states. The LCP has the potential to take exponential running time. Each wrong guess requires us to solve the linear system again, so even a moderate number of wrong guesses can kill real-time performance. Fortunately, the system has temporal coherence, and a good guess is to take the friction states from the previous time step. If these are wrong, we make local changes (driven by the constraints that are not satisfied) and usually find the correct states within a few trials. Pseudocode for our LCP solver appears in Section 3.6.1, which also discusses several other optimizations that accelerate this computation.

We set $f_i = f_{\text{perlen}}(l_{i-1} + l_i)/2$, where l_{i-1} and l_i are the rest lengths of the needle edges adjoining node i, and f_{perlen} is the needle-tissue friction per unit length. If node i is the needle tip and it is cutting, we add to f_i a cutting force f_{cut} that varies spatially according to the type of tissue the needle tip is currently cutting through. Simone and Okamura [2002] observed *capsule puncture*, a phenomenon where an unusually large force is required to cut the membrane surrounding an organ. We model it by setting f_{cut} to be large near a membrane, the prostate having a 0.5 mm thick capsule membrane and the skin being 1 mm thick.

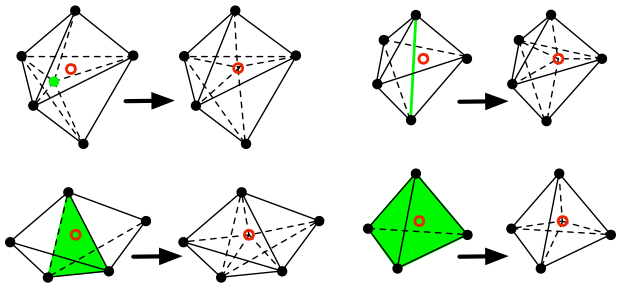

Figure 3: *Mesh operations: node snap, edge split (upper right), face split (lower left), tetrahedron split. The needle tip* $\mathbf{u}_{\text{new}}$ *is at the red circle. The simplex the operation acts on is green.*

Crouch et al. [2005] report that real-world needle-tissue friction is a function of their relative velocity. Rather than have f_i reflect this relationship, we instead include the velocity-induced friction in the force vector $\mathbf{F}(\cdot)$ in Equation (3). We apply an implicit viscous frictional force $f_i^{\text{vis}} = (\eta(l_{i-1} + l_i)/2)(\hat{\mathbf{v}}_i - \tilde{\mathbf{v}}_i)$ to needle node i (where η is a viscous damping coefficient) and an opposing force $-f_i^{\text{vis}}$ to tissue node i.

3.4 Remeshing and Reparameterization

After a simulated time step, some of the DYNAMIC coupling nodes of T and $\widetilde{T}$ are no longer coincident. However, our coupling method requires coupling nodes to have the same positions in both meshes. Thus, we dynamically adapt the meshes after each time step. Our mesh adaptation consists of two steps: needle tip remeshing and needle reparameterization.

3.4.1 Needle Tip Remeshing

When the needle tip node is DYNAMIC, we change the tissue mesh T, sometimes topologically, so that it has a node at the new location of the needle tip. Mesh changes occur only near the needle tip, and are quite inexpensive. We skip this step if the tip node is STATIC.

The goals of needle tip remeshing are to make T conform to the needle, to have tetrahedra of as high quality as possible, and to do so quickly. We remesh in material space by applying one of the candidate operations depicted in Figure 3: the node snap, the edge split, the face split, and the tetrahedron split.

Two ideas govern our remeshing algorithm. First, we choose among candidate operations by directly measuring the quality of the tetrahedra that would be created by each operation, and selecting the operation that maximizes the quality of the worst tetrahedron. Second, we maintain a stack of all the operations that have changed the mesh topology (*i.e.* all operations except the node snap), and we consider undoing the most recent operation before applying a new one. The stack is particularly important when the needle is retracted; our procedure is designed so that once the needle is fully withdrawn from the body, the tissue mesh will have returned to its original topology. We thereby prevent the accumulation of mesh quality degradation when the needle is inserted and withdrawn multiple times. Even when the needle is being inserted, the ability to undo the previous operation and replace it with a new one often offers better mesh quality.

For each candidate operation, we evaluate each new tetrahedron with a quality measure equal to its signed volume divided by the cube of its root-mean-squared edge length [Parthasarathy et al., 1994]. This measure is zero for a degenerate tetrahedron, and maximized by an equilateral tetrahedron. We have found this measure to

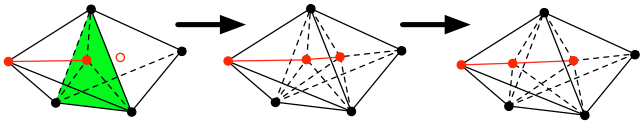

Figure 4: *The red edges and nodes are part of the needle shaft. The green face is split, creating a new needle tip node. We move the old needle tip node back along the needle shaft to maximize the mesh quality.*

be both a good reflection of a tetrahedron's fitness for finite element simulation and amenable to numerical optimization [Klingner and Shewchuk, 2007]. Tetrahedron quality is always computed from material (not world) coordinates. Let R be the mesh region comprising the union of all tetrahedra that can be deleted or changed by the candidate operations. The *quality vector* of the tetrahedra in R is a list of the tetrahedron qualities, sorted from worst to best. Our algorithm chooses the operation that lexicographically maximizes that quality vector (that is, it maximizes the worst tetrahedron, breaking ties by maximizing the second-worst, then the third-worst, etc.).

If the needle is penetrating tissue, we generate a set of candidate operations as follows. Let $\mathbf{u}_{\text{new}}$ be the new position of the needle tip in material space. (We obtain $\mathbf{u}_{\text{new}}$ by barycentric interpolation from the needle tip position in world space, and we obtain a tissue velocity and acceleration for it the same way.) We consider fifteen *standard operations* that transform the tetrahedron that contains $\mathbf{u}_{\text{new}}$: four node snaps, six edge splits, four face splits, and one tetrahedron split. Each operation places the new node or snapped node at $\mathbf{u}_{\text{new}}$. We also consider composite operations that first undo the operation atop the stack (if that operation created the needle tip node), then apply a new standard operation.

Candidate operations that would change the mesh boundary are discarded, except in the moment where the needle first penetrates the skin. Operations that fail to properly connect the needle nodes are also discarded. Let $\mathbf{u}_1$ and $\mathbf{u}_2$ be the old positions in material space of the needle tip and the needle node adjoining the tip, respectively. The needle nodes remain properly connected by the operations that snap $\mathbf{u}_1$ to $\mathbf{u}_{\text{new}}$, that create an edge connecting $\mathbf{u}_1$ to $\mathbf{u}_{\text{new}}$, or that delete $\mathbf{u}_1$ and create an edge connecting $\mathbf{u}_2$ to $\mathbf{u}_{\text{new}}$. (Undoing the top stack operation entails deleting $\mathbf{u}_1$ from both T and $\widetilde{T}$.)

Because the needle moves only a small distance during a time step, $\mathbf{u}_{\text{new}}$ tends to be close to $\mathbf{u}_1$ or $\mathbf{u}_2$, often producing a short edge that compromises the mesh quality. We avoid this pitfall by moving $\mathbf{u}_1$ to the optimal position on the segment $\mathbf{u}_{\text{new}}\mathbf{u}_2$, or (if the operation deletes $\mathbf{u}_1$) by moving $\mathbf{u}_2$ to the optimal position on the segment $\mathbf{u}_{\text{new}}\mathbf{u}_3$; see Figure 4. (To respect the needle curvature, we could have searched along the energy-minimizing curve that Spillmann and Teschner [2008] use, but we find that placing the node on the segment is good enough.) The "optimal" position is the one that maximizes the minimum quality among the tetrahedra that adjoin the moved node. This repositioning is part of the candidate operation, and is taken into account when the best operation is chosen.

Needle retraction uses somewhat different candidate operations. The *only* operation we consider that does not delete the needle tip $\mathbf{u}_1$ from the needle mesh is a node snap that moves $\mathbf{u}_1$ to $\mathbf{u}_{\text{new}}$. The other candidate operations delete $\mathbf{u}_1$ as follows. If $\mathbf{u}_1$ was created by the operation on top of the stack, then the stack is popped and that operation is undone, deleting $\mathbf{u}_1$ from both T and $\widetilde{T}$; otherwise, $\mathbf{u}_1$ was placed by a node snap, in which case we delete it from the needle mesh only. In either case, one of the standard operations subsequently creates a node at $\mathbf{u}_{\text{new}}$, or snaps a node there. If $\mathbf{u}_1$ survives in the tissue mesh, it tends to be close to $\mathbf{u}_{\text{new}}$, so we subsequently optimize the position of $\mathbf{u}_1$ (but not $\mathbf{u}_2$) as part of the candidate operation. Because $\mathbf{u}_1$ no longer lies on the needle, it can move freely.

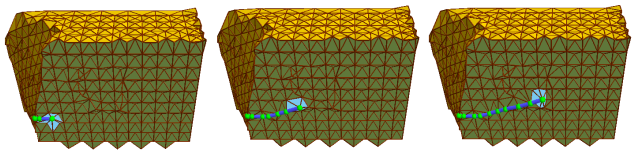

Figure 5: *Cutaway views of the changing mesh at several times during a simulation. The most recently changed tetrahedra (around the needle tip) are highlighted in blue.*

Our remeshing procedure is summarized in Algorithm 2. Figure 5 depicts mesh modifications during needle insertion. The remeshing algorithm relies on the presumption that the needle tip will not move much further than an element's width in a single time step. If we needed to accommodate larger needle movements in a single step, then the large motion could be broken down into smaller substeps with multiple applications of the inexpensive remeshing algorithm.

Algorithm 2 Tissue remeshing at the needle tip

```
/* Build set S of candidate operations */
S ← { node snap, moving needle tip u_1 to u_new}
t ← tetrahedron in T containing u_new
if s_1 = DYNAMIC+ (the needle is penetrating) then
    S ← S ∪ set of standard operations on t that place a node
    (other than u_1) at u_new that is connected to u_1, then optimize the position of u_1 constrained to lie on u_new u_2
if u_1 was created by the operation atop the stack then
    t_undo ← tetrahedron that will contain u_new if the operation atop the stack is undone
    S ← S ∪ set of operations that undo the top stack operation, then perform a standard operation on t_undo that places a node at u_new that is connected to (or is) u_2, then optimize the position of the needle node adjoining u_new
else if s_1 = DYNAMIC− (the needle is retracting) then
    S ← S ∪ set of standard operations on t that place a node at u_new that is connected to (or is) u_2, then freely optimize the position of u_1.
T' ← set of tetrahedra deleted/changed by operations in S
for each candidate operation o ∈ S do
    Compute the quality vector for the tetrahedra created by o and the tetrahedra in T' not deleted by o
Perform operation that maximizes the quality vector
if optimal operation undoes the top stack operation then
    Pop the stack
if optimal operation includes a face/edge/tetrahedron split then
    Push the split operation onto the stack
```

Because of the restricted demands of our application, our remesher is effective at maintaining element quality: in all the simulations we have run, after the first twenty time steps, no tetrahedron's dihedral angle has been smaller than 10.3° or larger than 160.0°. During the first few time steps after the needle punctures the skin, it is impossible to prevent the presence of a very short edge connecting the needle tip to the mesh surface, with flat tetrahedra adjoining it; but at all other times, tetrahedron quality is good. Table 2 in Section 4 shows that remeshing is cheap compared to the simulation.

3.4.2 Needle Reparameterization

Our needle tip remeshing procedure ensures that the tissue mesh has a sequence of nodes and edges that corresponds to the part of the needle inside the tissue. These nodes will be the coupling nodes in the next time step, and their positions are determined by the tissue mesh—that is, $\tilde{\mathbf{x}}_i^{k+1} = \hat{\mathbf{x}}_i^{k+1}$. The FREE nodes' positions are determined by the solution to Equation (7); that is, $\tilde{\mathbf{x}}_i^{k+1} = \tilde{\mathbf{x}}_i^{*k+1}$.

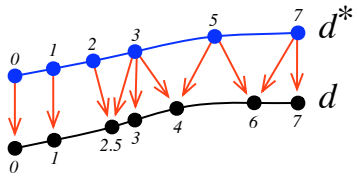

Figure 6: *Needle reparameterization. The numbers are d^* (top) and d (bottom). The red arrows indicate which old nodes and new nodes' quantities are related via piecewise linear interpolation.*

Needle sliding at a DYNAMIC node implies that it no longer represents the same point on the needle as it did before the time step. Moreover, remeshing creates and deletes nodes, and the simulation does not keep the needle perfectly inextensible, so the needle length varies slightly. Therefore, we reparametrize the needle and interpolate physical quantities from before to after the time step.

We parametrize each node i existing before the time step by its distance d_i^* from the base of the needle, and each node j existing after by its distance d_j from the base after the time step. (We compute these distances as sums of line segment lengths, but one could use the arc length of an interpolating curve instead.) Because the needle is not perfectly inextensible, we scale all the distances after the time step so the values of d^* and d at the needle tip are equal.

To compute the acceleration $\tilde{\mathbf{a}}_j$ at node j after a time step, we build an interpolating function $g(\cdot)$ such that $g(d_i^*) = \tilde{\mathbf{a}}_i^*$, where the right-hand side comes from the solution of Equation (7), then set $\tilde{\mathbf{a}}_j = g(d_j)$, as illustrated in Figure 6. We use Akima's interpolation [1970] to construct the interpolating functions for the parameters used in the needle force computations of Bergou et al. [2008] and Spillmann and Teschner [2007] such as the twist angle θ_i (the angle of deviation from the Bishop frame at node i) and the rest curvature $\bar{\omega}_i$. We treat the rest lengths l_j of the needle edges specially by first constructing $L_0^* = 0, L_i^* = \sum_1^i l_i^*$, then using Akima's interpolation for L, then setting $l_q = L_q - L_{q-1}$. We use piecewise linear interpolation for velocity and acceleration, as Akima's interpolation is not monotonic and could compromise stability.

A needle edge outside the tissue can become too short or too long in two places: where the needle exits the guide sleeve (see Section 3.5), and where the needle enters the tissue. Thus, we merge nodes that are too close together (shorter than half the minimum initial edge length), moving a FREE node onto the node on the surface of the tissue or the end of the sleeve; and we split edges that are too long (over four times the maximum initial edge length), all before reparameterizing. To split an edge, we place a new node at the midpoint of an interpolating cubic curve.

3.5 Bevel Tip and Needle Base Manipulation

The formulation above suffices to model a needle whose tip is symmetric, causing it to penetrate tissue in a straight line. However, surgeons can more easily circumvent obstacles by using bevel-tip needles that move on curved paths because the beveled tip compresses tissue asymmetrically. The compressed tissue exerts a force that bends the needle, as shown in Figure 7a. This phenomenon occurs at a scale too small for the tetrahedral mesh to simulate. Instead, we approximate the effect by adding a displacement to the needle tip material coordinate $\mathbf{u}_{\text{new}}$. The displacement added is along the second axis of the local coordinate frame of the needle tip, $\mathbf{m}_2$ as defined by Bergou et al. [2008]. The magnitude of the displacement is $h = d \tan\psi$, where d is the distance the tip moves along the tangential direction $\mathbf{t}_i$ during the current time step, and ψ is the bevel angle depicted in Figure 7b.

Steerable needles offer three control parameters to the surgeon: 1) v_{insert}, the speed at which the needle advances from the guide sleeve, 2) $\mathbf{v}_{\text{base}}$, the rigid-body velocity of the sleeve that holds the base of the needle, and 3) ϕ, the angular velocity of the needle base. The portion of the needle inside the sleeve moves with the sleeve, so we simulate only two nodes inside the sleeve, whose velocities are fixed at $\mathbf{v}_{\text{base}}$. We must constrain two nodes because the direction of the edge at the base of the needle is dictated by the sleeve orientation. A time step increases the length of the second edge (just outside the sleeve) by $v_{\text{insert}}\Delta t$, effectively pushing the needle out through the sleeve. To rotate the base, we add $\phi\Delta t$ to θ_i at the needle's base node i. The simulation propagates this rotation to the bevel tip.

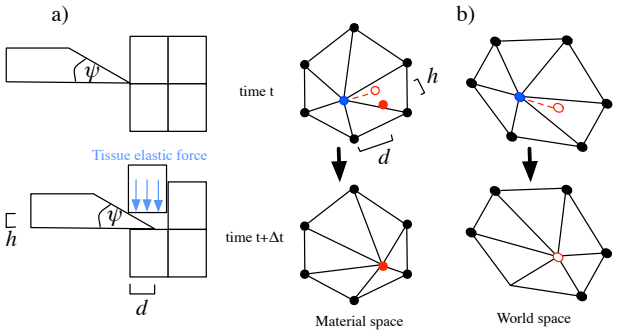

Figure 7: *a) As the needle penetrates the tissue, the bevel tip compresses the tissue in one direction more than the other, inducing an asymmetric elastic tissue force that causes the needle to bend. As the needle moves a distance d, it displaces the tissue on one side by a distance of $h = \tan\psi$. b) The current tip position appears as a blue circle. The new tip position appears as a hollow red circle. The new tip position is displaced to the solid red circle in material space, but not in world space. This displacement simulates a strain near the needle tip that will bend the needle in subsequent time steps.*

3.6 Optimizations

We have several optimizations that reduce the simulation's running time. These optimizations should be applicable to other types of simulation.

3.6.1 Accelerating the LCP Solver

The simulation's bottleneck is solving linear complementarity problems. Its speed depends crucially on rapidly determining the friction state variables s_i. Empirically, these values are temporally coherent if the needle is manipulated smoothly (as it should be for this application). The obvious strategy for guessing the states works well: presume that the state variables remain the same from time step to time step. However, needle reparametrization means that there is not necessarily a one-to-one mapping from nodes in one time step to nodes in another. We guess the state at each new node by using the nearest node (measured by arc length) from the previous time step.

After solving the coupled system (7) for a set of friction states, we check if the constraints (9), (10), and (11) are satisfied. If not, we try to alter the state of each node whose constraint is not satisfied using the following rules. If the node is DYNAMIC, change it to STATIC. If the node is STATIC, change it to DYNAMIC with its sign determined by the coupling constraint force $[1\ 0\ 0]^{\mathsf{T}} \cdot \mathbf{c}_i$.

To avoid trying the same configuration twice, we remember those we have tried. If we are about to try one we have already tried, we instead try the next configuration in lexicographic order, with the node at the tissue entry point acting as the least significant "digit."

With this heuristic, we find a consistent set of friction states in fewer than 1.2 trials on average in our examples. Temporal coherence is lost when the needle switches from cutting to withdrawing, in

which case many iterations might be required to find a consistent configuration. We devised two additional optimizations to keep the frame rate high during these transitions.

Our conjugate gradient solver is always initialized with the solution from the previous (unsuccessful) trial. This simple trick significantly cuts down the number of iterations the conjuage gradient method takes to converge to a tolerance τ_{sol} (by a factor of twenty in our prostate examples). Interestingly, we did not attain speedups by initializing the solver to the solution from the previous time step.

We took this optimization a step further by deciding whether to change state variables before the solver fully converges. As soon as the conjugate gradient residual drops below a tolerance τ_{check} (which is larger than τ_{sol}), we check if any constraints are violated, and switch state variables immediately if they are. In our simulation results, $\tau_{\text{check}} = 10^{-3}$ and $\tau_{\text{sol}} = 10^{-5}$.

Finally, if a consistent configuration isn't found within ten trials, we simply accept the last configuration to ensure real-time performance. In our prostate example, our solver found a consistent configuration within ten trials for 99.6% of the time steps. Our LCP solver is summarized in Algorithm 3.

Algorithm 3 Solve the coupled system, yielding the friction states and the tissue and needle accelerations

```
Guess friction states s_i from previous time step
Initialize solution to 0
for numTrials ← 0 to maxTrials do
    Solve linear system (7) to tolerance τ_check
    if the constraints (9), (10), and (11) are satisfied then
        Continue solving linear system to tolerance τ_sol
        if the constraints (9), (10), and (11) are satisfied then
            return solution
    Change friction states s_i to something new (see §3.6.1)
    /* Next iteration will reuse solution from this trial */
Solve linear system (7) to tolerance τ_sol
return (approximate) solution
```

3.6.2 Parallel Sparse Conjugate Gradients

We use the OpenMP conjugate gradient implementation, which parallelizes the sparse matrix multiplication and all the vector operations. To improve the cache performance of sparse matrix access, we order the tissue mesh nodes with a reverse Cuthill–McKee ordering [Cuthill and McKee, 1969], which puts the nonzero entries close to the diagonal. We avoid building a new sparse matrix for each solution by treating the coupling terms separately from the other terms in the matrix multiplication. In Equation (7), the matrices $\hat{\mathbf{A}}$ and $\tilde{\mathbf{A}}$ are fixed during an LCP solution, whereas $\mathbf{R}$ and $\mathbf{Z}$ vary with the coupling states. We never assemble the third row or column of (7); we perform that portion of the matrix-vector multiplication element by element.

We store $\hat{\mathbf{A}}$ with a dynamic compressed sparse row format that can be updated to reflect tissue remeshing without being reassembled from scratch. As tetrahedra are deleted, created, and modified, we update their contributions to $\hat{\mathbf{A}}$. Each compressed row of the matrix is stored in a separate array that can be resized to accommodate new edges. An edge deletion causes two zero entries to be written into the matrix. A node deletion causes an entire row and column to be set to zero. We maintain a list of free node indices so unused rows can be reassigned to newly inserted nodes. Unused column entries in an active row are set to zero. Because remeshing changes only a small portion of the mesh, the time consumed by these zero entries is negligible compared to rebuilding the matrix. The most common remeshing operation is to snap a node to the needle tip, which entails no change to the sparse matrix structure.

e_{thres}	% updated	F&J (ms)	% frame time	% rel error
0	100.00	12.42	100.00	0.00
100	7.96	3.83	82.37	0.06
1000	3.65	2.52	79.68	0.19
10000	0.58	2.12	78.89	0.16

Table 1: *For several values of e_{thres}, the % of tetrahedra whose Jacobian matrices are updated per time step, the time required for the force and Jacobian computations (F&J), the % of total frame time compared to recomputing every Jacobian, and the relative error in the tip position after four seconds of needle insertion (200 time steps). Values are for the prostate mesh (~13,375 tetrahedra) running with seven threads on an 8-core processor.*

3.6.3 Parallel Incremental Jacobian Update

We also parallelize the computation of the forces and the Jacobian matrices. To reduce lock contention, we partition the mesh with METIS [Karypis and Kumar, 1998], so that the entries modified by one thread do not overlap much with those of any other.

We observe that the Jacobians tend to change slowly over time, because most tetrahedra deform little during a time step. Therefore, we only update a tetrahedron's Jacobian matrix's contributions to $\hat{\mathbf{A}}$ (as determined by Equation (4)) when it has changed enough since its last update. The test is whether

$$e = \left(\max_{i,j} |\triangle U_{ij}| \max_{i,j} |\triangle V_{ij}| \right)^2 \max_{i,j} |N_{ij}| \max_{i,j} |B_{ij}| \max(\lambda, \alpha) \qquad (12)$$

exceeds a threshold e_{thres}, where U and V are the SVD rotation matrices mentioned in Section 3.2, N is a matrix whose columns are area-weighted normal vectors to the tetrahedron faces, and B is the tetrahedron's barycentric matrix. λ and α are the second coefficients for computing the Piola–Kirchhoff stress and the damping stress in Irving et al. [2004], respectively. e is an approximate upper bound on the change in any entry of the Jacobian matrix due to changes in U and V, for both the elastic and damping forces. Table 1 shows the running time reductions and the relative errors in needle tip position caused by lazy updating for several values of e_{thres}. Lazy updating reduces the tissue force and Jacobian computation time by 83% and the frame time by 20%, with no more than a 0.2% relative error in the needle tip position.

4 Results and Discussion

All timings in this paper are on an 8-core 3.0 GHz Intel Xeon with 16 GB RAM. Figure 1 illustrates our simulations of a prostate cancer brachytherapy procedure, wherein a needle implants radioactive seeds in the prostrate gland, for both a flexible bevel-tip needle (a) and a stiff symmetric-tip needle (b). We use isosurface stuffing [Labelle and Shewchuk, 2007] to generate a tetrahedral mesh encompassing an anatomically accurate model of the epidermis, dermis, hypodermis, urethra, prostate, perineum, pelvic bone, bulbourethral glands, vasa deferentia, seminal vesicles, and bladder. The external mesh boundary conforms to the skin, and internal triangular faces conform to the prostate boundary, separating regions with dissimilar material properties. Material parameters for prostate and other tissues are inferred from Krouskop et al. [1998]. The bone is constrained not to move. A video accompanying this paper demonstrates the needle being inserted into and removed from the tissue, both slowly and vigorously, to demonstrate the simulator's stability.

We also constructed an artificial example, shown in Figure 8, where we simulate the ability of the bevel-tip flexible needle to be steered by rotating its base to avoid fixed cylindrical obstacles and hit a target in a deformable tissue.

To evaluate our simulator, we compare the simulated insertion of steerable needles into gel tissue phantoms with real-world experiments performed at Johns Hopkins University in 2005. In these

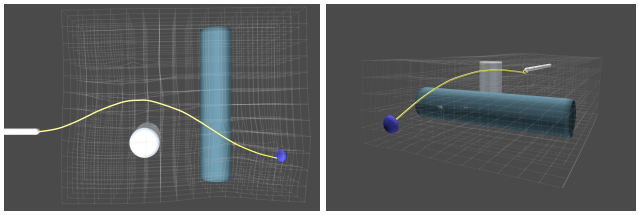

Figure 8: *An example showing the bevel-tip flexible needle steering to avoid two fixed cylindrical obstacles. The base of the needle is rotated to change the direction of bending.*

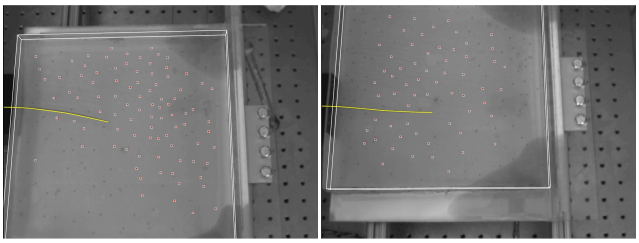

Figure 9: *Comparison of our simulation with two experiments. The simulated needle appears in blue. The red squares are the physical markers, and the yellow dots are the simulated markers. Our root-mean-squared simulation error (compared with the experiment) is* 0.75 *mm for the* single-bend *experiment (left), and* 1.30 *mm for the* double-bend *experiment (right). The* double-bend *experiment used the* single-bend *parameters without re-tuning.*

experiments, a robotic device [Webster III et al., 2005a] inserts a bevel-tip flexible needle of diameter 0.83 mm into a 27.1×26.5×3.9 cm gelatin block. Fiducial markers are placed on the surface of the gel phantom to track the deformations. The material's Young's modulus (E) and Poisson's ratio (ν) were measured using compression tests. We use these values in our simulation.

The remaining simulation parameters (tissue damping, needle parameters) were tuned by hand to match a *single-bend* experiment, where the needle is inserted and retracted from the tissue without twisting. The same parameters are then used without re-tuning in the second *double-bend* experiment, where the needle is twisted 180° halfway through insertion. Figure 9 shows screenshots from both experiments.

We ran the simulation, tracked the positions of simulated markers as computed by barycentric interpolation on the tetrahedra, and compared their positions to the physical markers in the experiment. The trajectory of the real needle and the simulated needle match to video resolution for both experiments. The root-mean-squared error of the marker positions over time in the single bend experiment is 0.75 mm, with 88.3% of errors under 1 mm, and 97.8% of errors under 2 mm. The root-mean-squared error in the double bend example is 1.3 mm, with 90.5% of errors under 2 mm, and 97.2% of errors under 3 mm. The simulation also matches the experiments qualitatively, as the companion video shows.

Some of the marker error is attributable to the vision algorithm. It fails to track several markers (which we exclude from the videos), and several others are tracked incorrectly so they appear to jitter about.

Table 2 gives the running times for the different simulations discussed in this section. Observe that speeds for the steerable needle simulation in prostate tissue range from about 7 frames per second on one core to about 25 frames per second on seven cores. The LCP solver dominates the running time, whereas the local remeshing step takes a negligible amount of time. Table 3 shows the number of LCP trials per time step and the number of conjugate gradient iterations per linear solution for each example.

Name	#	Total	LCP	Tissue	Needle	Remesh
ProsFB	1	130.9	108.8	13.4	1.3	0.5
ProsFB	2	77.7	62.3	7.7	1.6	0.5
ProsFB	3	64.1	51.1	4.7	1.4	0.4
ProsFB	4	56.6	44.8	3.9	1.4	0.7
ProsFB	5	47.6	36.7	3.3	1.8	0.4
ProsFB	6	39.6	28.8	3.0	1.4	0.5
ProsFB	7	38.5	28.3	2.2	1.4	0.3
ProsSS	7	38.2	28.7	2.1	1.1	0.5
TwoCyl	7	24.0	14.4	1.9	0.6	0.9
1bend	7	22.8	13.2	1.1	0.4	0.9
2bends	7	33.0	23.6	1.1	0.5	0.5

Table 2: *Timings (ms) for several examples with different numbers of threads (#). Examples include the prostate mesh with flexible bevel-tip needle (ProsFB) or stiff symmetric-tip needle (ProsSS), the two-cylinder example (TwoCyl), and the tissue phantom verification experiments (1bend and 2bends). The frame time (Total) is divided into the LCP solution (LCP), the force and Jacobian computations for tissue (Tissue) and needle (Needle), and remeshing and needle reparameterization (Remesh). The number of tetrahedra/vertices for prostate, two cylinders, and tissue phantoms are 13,375/2,763, 3,248/813, and 2,280/672, respectively.*

Name	LCP	CG
ProsFB	1 / 1.41 / 10	18 / 205 / 652
ProsSS	1 / 1.23 / 4	12 / 225 / 536
TwoCyl	1 / 1.02 / 2	42 / 213 / 474
1bend	1 / 1.17 / 4	46 / 234 / 343
2bends	1 / 1.04 / 3	54 / 413 / 686

Table 3: *Minimum/average/maximum number of trials required for the LCP solver and number of conjugate gradient (CG) iterations for each linear solution for examples running on seven threads.*

5 Conclusions and Future Work

Our coupled needle-tissue simulator is capable of interactive and accurate simulation of a range of needles, from a symmetric-tip stiff needle to a bevel-tip flexible needle. We achieve this with a novel remeshing algorithm, a needle-tissue coupling algorithm, optimization techniques to improve LCP solution time and accelerate Jacobian computation, and parallelization.

We plan to use the simulator in surgical planning. Concurrent to this paper, members of our project group have developed feedback control software that uses our simulator to maneuver a steerable needle to a target in a deformable tissue [Hauser et al., 2009]. Figure 10 shows several states during needle insertion. The light blue helical paths are trajectories chosen by the planner at different instants in time. As the needle is inserted, our simulator predicts how the tissue deforms, and the planner uses this feedback to make trajectory corrections. Our future plans are to integrate into our planner the ability to avoid obstacles and to achieve coverage of a prostate gland with radioactive seeds with as few needle insertions as possible.

We are also exploring ways to extend the versatility of our simulator. The biomedical community is considering needles with prebent tips, giving them a much smaller turning radius [Reed et al., 2008]. Because the bend is not at the needle tip, simulating the needle entails placing a mesh node at a fixed location in the needle that moves with the needle. This necessitates a change to our remeshing algorithm that makes it difficult to maintain high mesh quality. A second extension would be to use graded meshes, which would allow us to model a larger tissue yet have higher resolution near the needle. The difficulty is to maintain the quality of the mesh during local remeshing. A third extension would be to model the

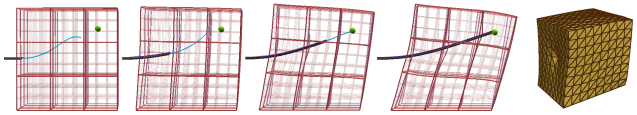

Figure 10: *A planning algorithm finds a path by which a steerable needle can reach a target using the simulator.*

torsional friction that couples a twisting needle to the surrounding tissue. Currently, we treat only sliding friction. To treat torsional friction as well, we would have to reformulate the force computation and take torque into account. A fourth extension would be to extend the remeshing to support the simulation of a procedure in which several needles are inserted to fix the tissue in place. Our simulator can already handle simulating multiple needles that never touch common elements or are inserted one at a time and retracted in the reverse order. More general uses would require a more sophisticated remeshing algorithm. The techniques reported here may also apply to simulating surgical sutures and staples where similar one-dimensional structures penetrate into soft tissue.

Acknowledgments

We thank the other members of the Berkeley Graphics Group and the Automation Lab for their helpful criticism and comments. We thank Allison Okamura, Noah Cowan, Robert Webster III, Landon Unninayar, and Vinutha Kallem at Johns Hopkins for the tissue phantom video and material parameters, Judy Hoffman for her marker tracking code, James Demmel and Samuel Webb Williams for discussions about sparse matrix solvers, Vincent Duindam of Intuitive Surgical and Jean Pouliot, I-Chow Joe Hsu, and Adam Cunha of UCSF for advice on physiology and prostate treatment. This work was supported in part by California MICRO 08-077, NSF Award CCF-0635381, NIH Award 1R01EB-006435-01A1, the UC Lab Fees Research Program, and by generous support from NVIDIA corporation, Pixar Animation Studios, Autodesk, Intel Corporation, and Sony Computer Entertainment America.

References

Abolhassani, N., Patel, R. V., and Moallem, M. 2007. Needle insertion into soft tissue: A survey. *Medical Engineering & Physics 29*, 4 (May), 413–431.

Akima, H. 1970. A new method of interpolation and smooth curve fitting based on local procedures. *J. ACM 17*, 4 (Oct.), 589–602.

Allard, J., Cotin, S., Faure, F., Bensoussan, P.-J., Poyer, F., Duriez, C., Delingette, H., and Grisoni, L. 2007. SOFA—An open source framework for medical simulation. In *Medicine Meets Virtual Reality 15*, IOS Press, 13–18.

Alterovitz, R., and Goldberg, K. 2008. *Motion Planning in Medicine: Optimization and Simulation Algorithms for Image-Guided Procedures*, vol. 50 of *Springer Tracts in Advanced Robotics*. Springer, Berlin, Germany.

Alterovitz, R., Pouliot, J., Taschereau, R., Hsu, I.-C., and Goldberg, K. 2003. Simulating needle insertion and radioactive seed implantation for prostate brachytherapy. In *Medicine Meets Virtual Reality 11*, IOS Press, 19–25.

Alterovitz, R., Goldberg, K., and Okamura, A. M. 2005. Planning for steerable bevel-tip needle insertion through 2D soft tissue with obstacles. In *IEEE International Conference on Robotics and Automation*, 1652–1657.

Alterovitz, R., Siméon, T., and Goldberg, K. 2007. The stochastic motion roadmap: A sampling framework for planning with Markov motion uncertainty. In *Robotics: Science and Systems III*, 233–241.

Bargteil, A. W., Wojtan, C., Hodgins, J. K., and Turk, G. 2007. A finite element method for animating large viscoplastic flow. *ACM Transactions on Graphics 26*, 3 (July), 16:1–16:8.

Bergou, M., Wardetzky, M., Robinson, S., Audoly, B., and Grinspun, E. 2008. Discrete elastic rods. *ACM Transactions on Graphics 27*, 3 (Aug.), 63:1–63:12.

Bertails, F., Audoly, B., Cani, M.-P., Querleux, B., Leroy, F., and Lévêque, J.-L. 2006. Super-helices for predicting the dynamics of natural hair. *ACM Transaction on Graphics 25*, 3 (July), 1180–1187.

Bertails, F. 2009. Linear time super-helices. *Computer Graphics Forum 28*, 2 (Apr.), 417–426.

Çavuşoğlu, M. C., Göktekin, T. G., and Tendick, F. 2006. GiPSi: A framework for open source/open architecture software development for organ level surgical simulation. *IEEE Transactions on Information Technology in Biomedicine 10*, 2 (Apr.), 312–322.

Crouch, J. R., Schneider, C. M., Wainer, J., and Okamura, A. M. 2005. A velocity-dependent model for needle insertion in soft tissue. In *Medical Image Computing and Computer-Assisted Intervention*, vol. 3749 of *LNCS*. Springer, Berlin, Oct., 624–632.

Cuthill, E., and McKee, J. 1969. Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 24th National Conference*, ACM, New York, 157–172.

Debunne, G., Desbrun, M., Cani, M.-P., and Barr, A. 2000. Adaptive simulation of soft bodies in real-time. In *Computer Animation 2000*, 15–20.

Dehghan, E., and Salcudean, S. E. 2007. Needle insertion point and orientation optimization in non-linear tissue with application to brachytherapy. In *2007 IEEE International Conference on Robotics and Automation*, 2267–2272.

DiMaio, S. P., and Salcudean, S. E. 2005. Needle steering and motion planning in soft tissues. *IEEE Transactions on Biomedical Engineering 52*, 6 (June), 965–974.

Gallagher, A. G., Ritter, E. M., Champion, H., Higgins, G., Fried, M. P., Moses, G., Smith, C. D., and Satava, R. M. 2005. Virtual reality simulation for the operating room: Proficiency-based training as a paradigm shift in surgical skills training. *Annals of Surgery 241*, 2 (Feb.), 364–372.

Goksel, O., Salcudean, S. E., and DiMaio, S. P. 2006. 3D simulation of needle-tissue interaction with application to prostate brachytherapy. *Computer Aided Surgery 11*, 6 (Nov.), 279–288.

Goktekin, T. G., Bargteil, A. W., and O'Brien, J. F. 2004. A method for animating viscoelastic fluids. *ACM Transactions on Graphics 23*, 3 (Aug.), 463–467.

Grégoire, M., and Schömer, E. 2006. Interactive simulation of one-dimensional flexible parts. *Proceedings of the 2006 Symposium on Solid and Physical Modeling*, 95–103 (June).

Hauser, K., Alterovitz, R., Chentanez, N., Okamura, A., and Goldberg, K. 2009. Feedback control for steering needles through 3D deformable tissue using helical paths. In *Robotics: Science and Systems V*.

Irving, G., Teran, J., and Fedkiw, R. 2004. Invertible finite elements for robust simulation of large deformation. In *Proceedings of the 2004 Symposium on Computer Animation*, 131–140.

Irving, G., Schroeder, C., and Fedkiw, R. 2007. Volume conserving finite element simulations of deformable models. *ACM Transactions on Graphics 26*, 3, 13:1–13:6.

Karypis, G., and Kumar, V. 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing 20*, 1 (Aug.), 359–392.

KLINGNER, B. M., AND SHEWCHUK, J. R. 2007. Aggressive tetrahedral mesh improvement. In *Proceedings of the 16th International Meshing Roundtable*, 3–23.

KOHN, L. T., CORRIGAN, J. M., AND DONALDSON, M. S. 2000. *To Err Is Human: Building a Safer Health System*. New York: National Academy.

KROUSKOP, T. A., WHEELER, T. M., KALLEL, F., GARRIA, B. S., AND HALL, T. 1998. Elastic moduli of breast and prostate tissues under compression. *Ultrasonic Imaging 20*, 4 (Oct.), 260–274.

LABELLE, F., AND SHEWCHUK, J. R. 2007. Isosurface stuffing: Fast tetrahedral meshes with good dihedral angles. *ACM Transactions on Graphics 26*, 3 (July), 57:1–57:10.

LINDBLAD, A., AND TURKIYYAH, G. 2007. A physically-based framework for real-time haptic cutting and interaction with 3D continuum models. In *Proceedings of the 2007 Symposium on Solid and Physical Modeling*, ACM, New York, 421–429.

LOOCK, A., AND SCHÖMER, E. 2001. A virtual environment for interactive assembly simulation: From rigid bodies to deformable cables. In *5th World Multiconference on Systemics, Cybernetics and Informatics*, 325–332.

MARCHAL, M., PROMAYON, E., AND TROCCAZ, J. 2006. Simulating prostate surgical procedures with a discrete soft tissue model. In *Third Eurographics Workshop in Virtual Reality Interactions and Physical Simulations*, 109–118.

MARCIA, R. F. 2008. On solving sparse symmetric linear systems whose definiteness is unknown. *Applied Numerical Mathematics 58*, 4 (Apr.), 449–458.

MENDOZA, C., AND LAUGIER, C. 2003. Simulating soft tissue cutting using finite element models. In *2003 IEEE International Conference on Robotics and Automation*, 1109–1114.

MÜLLER, M., AND GROSS, M. H. 2004. Interactive virtual materials. In *Graphics Interface 2004*, 239–246.

MÜLLER, M., DORSEY, J., MCMILLAN, L., JAGNOW, R., AND CUTLER, B. 2002. Stable real-time deformations. In *Proceedings of the 2002 Symposium on Computer Animation*, ACM, New York, 49–54.

NIENHUYS, H.-W., AND VAN DER STAPPEN, A. F. 2001. A surgery simulation supporting cuts and finite element deformation. In *Medical Image Computing and Computer-Assisted Intervention, 4th International Conference*, 153–160.

NIENHUYS, H.-W., AND VAN DER STAPPEN, A. 2004. A computational technique for interactive needle insertions in 3D nonlinear material. In *IEEE International Conference on Robotics and Automation*, vol. 2, 2061–2067.

O'BRIEN, J. F., AND HODGINS, J. K. 1999. Graphical modeling and animation of brittle fracture. In *Computer Graphics (SIGGRAPH '99 Proceedings)*, ACM Press, New York, 137–146.

O'BRIEN, J. F., BARGTEIL, A. W., AND HODGINS, J. K. 2002. Graphical modeling and animation of ductile fracture. In *Computer Graphics (SIGGRAPH 2002 Proceedings)*, 291–294.

PAI, D. K. 2002. STRANDS: Interactive simulation of thin solids using Cosserat models. *Computer Graphics Forum 21*, 3 (Sept.), 347–352.

PAIGE, C. C., AND SAUNDERS, M. A. 1975. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis 12*, 4 (Sept.), 617–629.

PARTHASARATHY, V. N., GRAICHEN, C. M., AND HATHAWAY, A. F. 1994. A comparison of tetrahedron quality measures. *Finite Elements in Analysis and Design 15*, 3 (Jan.), 255–261.

PICINBONO, G., DELINGETTE, H., AND AYACHE, N. 2003. Non-linear anisotropic elasticity for real-time surgery simulation. *Graphical Models 65*, 5 (Sept.), 305–321.

REED, K. B., KALLEM, V., ALTEROVITZ, R., GOLDBERG, K., OKAMURA, A. M., AND COWAN, N. J. 2008. Integrated planning and image-guided control for planar needle steering. In *Proceedings of the Second IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, 819–824.

SATAVA, R. M. 2005. Identification and reduction of surgical error using simulation. *Minimally Invasive Therapy & Allied Technologies 14*, 4–5 (Sept.), 257–261.

SEYMOUR, N. E., GALLAGHER, A. G., ROMAN, S. A., O'BRIEN, M. K., BANSAL, V. K., ANDERSEN, D. K., AND SATAVA, R. M. 2002. Virtual reality training improves operating room performance: Results of a randomized, double-blinded study. *Annals of Surgery 236*, 4 (Oct.), 458–463.

SIFAKIS, E., SHINAR, T., IRVING, G., AND FEDKIW, R. 2007. Hybrid simulation of deformable solids. In *Proceedings of the 2007 Symposium on Computer Animation*, 81–90.

SIMONE, C., AND OKAMURA, A. M. 2002. Modeling of needle insertion forces for robot-assisted percutaneous therapy. In *IEEE International Conference on Robotics and Automation*, 2085–2091.

SPILLMANN, J., AND TESCHNER, M. 2007. CORDE: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects. In *Proceedings of the 2007 Symposium on Computer Animation*, Eurographics Association, 63–72.

SPILLMANN, J., AND TESCHNER, M. 2008. An adaptive contact model for the robust simulation of knots. *Computer Graphics Forum 27*, 2 (April), 497–506.

TASCHEREAU, R., POULIOT, J., ROY, J., AND TREMBLAY, D. 2000. Seed misplacement and stabilizing needles in transperineal permanent prostate implants. *Radiotherapy and Oncology 55*, 1 (Apr.), 59–63.

TAYLOR, R. H. 2006. A perspective on medical robotics. *Proceedings of the IEEE 94*, 9 (Sept.), 1652–1664.

TERZOPOULOS, D., PLATT, J., BARR, A., AND FLEISCHER, K. 1987. Elastically deformable models. In *Computer Graphics (SIGGRAPH '87 Proceedings)*, 205–214.

VIDAL, F. P., JOHN, N. W., HEALEY, A. E., AND GOULD, D. A. 2008. Simulation of ultrasound guided needle puncture using patient specific data with 3D textures and volume haptics. *Computer Animation and Virtual Worlds 19*, 2 (May), 111–127.

WANG, X., AND FENSTER, A. 2004. A virtual reality based 3D real-time interactive brachytherapy simulation of needle insertion and seed implantation. In *2004 IEEE International Symposium on Biomedical Imaging*, 280–283.

WEBSTER III, R. J., MEMISEVIC, J., AND OKAMURA, A. M. 2005. Design considerations for robotic needle steering. In *2005 IEEE International Conference on Robotics and Automation*, 3588–3594.

WEBSTER III, R. J., OKAMURA, A. M., COWAN, N. J., CHIRIKJIAN, G. S., GOLDBERG, K., AND ALTEROVITZ, R., 2005. Distal bevel-tip needle control device and algorithm. U.S. patent application number 11/436,995, May.

WEBSTER III, R. J., KIM, J. S., COWAN, N. J., CHIRIKJIAN, G. S., AND OKAMURA, A. M. 2006. Nonholonomic modeling of needle steering. *Int. Journal of Robotics Research 25*, 5–6 (May), 509–525.

WOJTAN, C., AND TURK, G. 2008. Fast viscoelastic behavior with thin features. *ACM Transactions on Graphics 27*, 3 (Aug.), 47:1–47:8.