

A real time finite element based tissue simulation method incorporating nonlinear elastic behavior

HUALIANG ZHONG*, MARK P. WACHOWIAK and TERRY M. PETERS

Imaging Research Laboratories, Robarts Research Institute, 100 Perth Drive, London, ON, Canada, N6A 5K8

(Received 21 January 2005; revised 9 June 2005; in final form 2 August 2005)

This paper presents a new finite element simulation approach for surgical simulators. Based on the solution of the algebraic equations derived from a nonlinear elastic model, we propose a real time simulation rule based on the implicit relation between the displacements of contacted and free nodes. This rule is an analytic expression in the linear case, and an approximation of the implicit relation in the non-linear case. We also remove some of the restrictions on flexibility exhibited by previous linear and nonlinear approaches. In the linear case, real time reconfiguration of the contacted nodes and the boundary constraints is realized using the simulation rule, while in the nonlinear case, a similar result is obtained by employing affine mapping.

These methods allow nonlinear material properties to be applied to real time tissue simulation, with an efficiency comparable to that of the tensor matrix method for linear elastic models.

Keywords: Surgical simulation; Finite element method; Nonlinear elasticity; Tissue modeling

Summary of notation

Ω	continuous body in a three-dimensional space	S	set of contacted nodes in preprocessing stage
$U(u, v, w)$	vector of displacements	T	set of contacted nodes in animation time
\mathbf{f}	vector of external forces	$R_i^{F(P_K, P_B, P_R)}$	a DD-function related to the configuration $\Gamma(P_K, P_B, P_R)$, or simply by R_i
W	strain energy density function	\tilde{R}_i	polynomial approximation of R_i
Π_{ext}	external potential work		
C	right Cauchy-Green strain tensor		
G (or g_{ij})	left Cauchy-Green strain tensor		
E	Green strain tensor, defined by $E = (C - I)/2$		
σ	Cauchy stress		
S_{IJ}	the second Piola-Kirchhoff stress		
$\zeta(\zeta, \eta, \xi)$	local coordinate system		
$N_i(\zeta, \eta, \xi)$	natural shape basis function		
ω_i	domain in the i -th element under the current configuration		
K	global stiffness matrix		
\tilde{K}	reduced global stiffness matrix		
(P_K, P_B, P_R)	contacted nodes, fixed boundary nodes and other free nodes, respectively		

1. Introduction

Along with the development of imaging technologies, significant progress has been made in understanding the mechanical behavior of human tissue, enabling the development of surgical simulators. A surgical simulator must combine the visual realism provided by medical images with the organ's biomechanical properties. Since this information is not intrinsic to the images, a separate system that simulates the mechanical properties must be linked to the visual display so that the behavior of the images during the simulated intervention accurately reflects that of an actual organ. The concept of surgery simulation has evolved from simple shape and surface modeling, through the integration of haptics, and finally to

*Corresponding author. E-mail: hzhong@imaging.robarts.ca

physiological modeling and analysis (Delingette 1998, Bona *et al.* 2003, Pathmanathan *et al.* 2004). The increasing sophistication of such simulation systems has raised expectations relating to both visual realism and physical fidelity. Consequently, various deformable modeling approaches such as chain-mail, mass-spring models, or finite element methods (FEM) have been developed to meet these requirements (Lee *et al.* 1995, Gibson 1997, Baraff and Witkin 1998, Terzopoulos and Fleischer 1988, Terzopoulos and Waters 1990).

Finite-element methods commonly employed in continuum mechanics are an industry standard for prototype design, and their feasibility and accuracy have been proven in many fields of application (ABAQUS, Zienkiewicz and Taylor 2000, Fung 1981). Even though organs usually have complex micro structures that may result in various elastic properties such as non-linearity, anisotropy, creep or visco-elasticity, remarkable progress has been achieved in understanding their mechanic structures during the last several decades, especially with FEM approaches (Fung 1981, Ogden 1984, Glass *et al.* 1991). Motivated by the need to build real time surgical simulators, many researchers have concentrated on improving the computational efficiency of FEM systems.

1.1 Overview of real time finite element methods

With Terzopoulos' pioneering work in modeling elastically deformable objects, FEMs have been an attractive approach to visualization in the virtual reality domain (Terzopoulos and Fleischer 1988). However, as a modeling technique for surgery simulation, the speed of FEMs has not yet achieved real time performance (e.g. 20 frames/s) even for a linear model of 3000 nodes. In response to this limitation, several real time FE techniques have recently emerged.

Bro-Nielsen and Cotin (1996) separates all the nodes in the FE mesh into those on the surface and those in the interior. For the interior, a condensation approach is employed, based on an analytical relationship between the displacements of the internal nodes and those of the surface nodes. This procedure can greatly reduce the size of the stiffness matrix, and consequently, relieves the real time computational burden. Unfortunately, this approach is limited to linear elastic models. In his demonstration, the simulated force was applied to only a single node and haptic feedback was not supported (Berkley *et al.* 2004).

Later, Cotin *et al.* (1999) introduced a preprocessing concept that creates a tensor for each free node, so that in real time animations, the displacements of the contacted nodes can easily be used to generate the displacement of each free node. In Cotin's method, the tensor matrix is created by running a FE solver multiple times, with the generated displacements of each node then being used to build the matrix elements. This approach is based on a linear elastic model and can achieve the same accuracy as

a linear FE solver. However, it is not applicable to the case where the underlying topology of the mesh changes, as is the case in a surgical incision simulation. Likewise, in a realistic animation, with this approach, the external forces can only be applied at the same positions as those employed in the preprocessing stage.

Dynamic modeling, which decouples the consistent movements of all nodes by lumping a mass at each node, is another popular approach (Zienkiewicz and Taylor 2000). Consequently, the continuum object, which in full finite element analysis is interpreted as a large consistent system, is now simplified into a set of independent differential equations for each vertex, with each differential equation being solved through explicit or implicit integration methods. Because explicit integration can avoid the expenses of inverting the global stiffness matrix, dynamic modeling can be achieved at an acceptable refresh rate, but at the expense of accuracy and global consistency.

James and Pai (1999) employed a boundary element method (BEM) to model a deformable object based on linear elasticity. This approach also permits changes in boundary constraints in real time simulations, based on the use of the Woodbury formula (Hager 1989). However, the drawback of BEM is that it cannot be applied to inhomogeneous materials, and its speed is still limited.

A less restricted and more promising approach is the conjugate gradient method which Nienhuys and van der Stappen (2000) applied into the linear elastic model, based on the sparse property of the stiffness matrix. In this method, the complexity of the evaluation function is almost linear with respect to the number of all the nodes in the mesh, and its overall performance is subject to the number of iterations required for convergence. This model will become more feasible with increases in computer speed.

With these advanced real time FE techniques, deformable objects can be simulated with high visual and physical realism in a reasonable length of time. For example, Berkley *et al.* (2004) applied the condensation approach to simulate the virtual surgical suturing of a human hand model; Picinbono *et al.* (2001) applied the non-linear and anisotropic properties in a simulation of laparoscopic liver surgery based on the dynamic tensor-mass model (Cotin *et al.* 2000); with dynamic FE modeling, Zhuang and Canny (2000) realized geometrically non-linear deformation with haptic interaction and Wu *et al.* (2001) combined the dynamic FE approach with the Mooney-Rivlin elastic model.

Recently, Pathmanathan *et al.* (2004) showed that using the full nonlinear finite element formulation in a real time application is computationally feasible. They employed 4608 hexahedral elements (which have better convergent behavior than tetrahedra (Samani *et al.* 1999)) to simulate large deformations of the breast to predict tumor location. While this application demonstrates a new approach to computing the deformation of

soft tissue, it again raises the question of real-time computational performance.

For the full non-linear FEM approach, the stiffness matrix is no longer constant, so few applications can achieve a reasonable speed. Nienhuys (2003) embedded the localized conjugate gradient algorithm into Newton-Raphson method, but the overall speed of this method is at least 10 times slower than that for the linear FE model.

1.2 New approaches

Before introducing our work, we summarize the notations employed in the remainder of the paper. We define contacted nodes P_K to be those where probes or other surgical tools touch the organ's surface. Fixed boundary nodes P_B are those which have zero displacements in any direction throughout a simulation. Both P_K and P_B during a real-time animation time could be different from those in the preprocessing stage. Free nodes P_R are the nodes within, or on the boundary of the domain that have no external forces or prescribed constraints applied to them. The direct relationship between the displacements of contacted and free nodes is called the DD relation and mathematically expressed as the DD-functions.

Our contributions to this work are in the following areas: deriving an explicit expression of the DD-functions for linear models, interpolating these functions for a nonlinear elastic model and extending the flexibility of the linear and nonlinear real time approaches. In the linear case, our formula is different from the expression $u_i = K_{ii}^{-1}(f_i - K_{is}u_s)$ used in the condensation technique (Bro-Nielsen and Cotin 1996), where K_{ii}^{-1} must be recomputed at significant time cost if the contacted or the fixed boundary nodes are changed. Our expression in the linear case is more flexible than those presented by Bro-Nielsen and Cotin (1996) and Cotin *et al.* (1999). The full nonlinear finite element solver, to our knowledge, has not yet been employed to build a surgery simulator. Since this is partly due to its speed, we propose an interpolation approach to the DD-functions to permit real time performance. The following sections consist principally of two parts: (1) we outline the formulation procedure of the non-linear FEM, describe the computing technique we employed and then validate the established solver; (2) from the formulation procedure, we derive the concept of the DD-functions, give their expression for a linear model, interpolate these functions with the above solver for a nonlinear model, and then introduce an affine mapping for real time simulation approaches.

2. Implementation of non-linear finite element method

Since the elastic response of soft tissue is typically non-linear (see figure 1a), classic linear FE models cannot precisely characterize anatomical behavior or provide

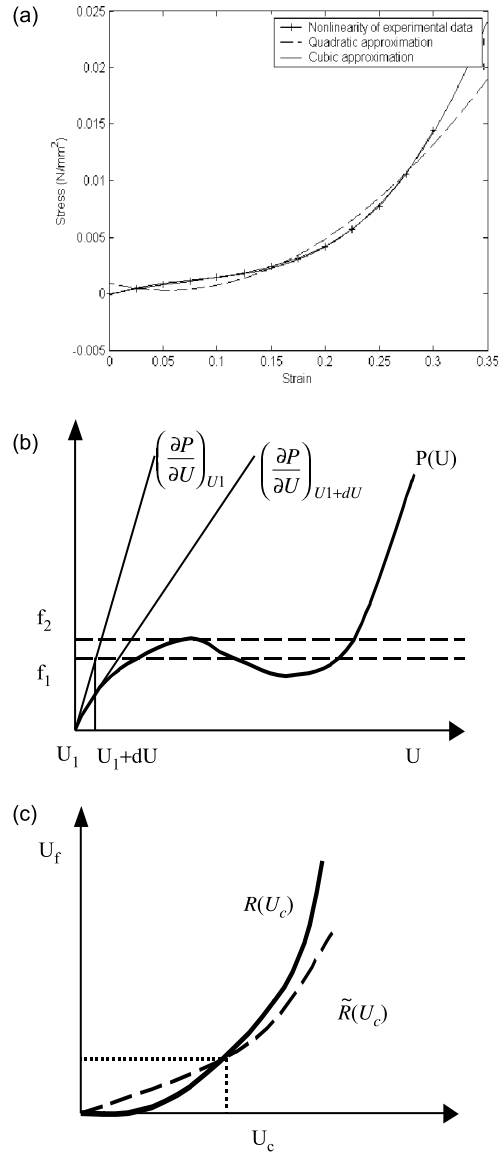


Figure 1. (a) An example of nonlinearity between stress and strain and its quadratic and cubic approximations. (b) Newton's method to find the solution of $P(U) = \mathbf{f}$ describing the relation between force \mathbf{f} and displacement U . (c) The displacement U_f of the free nodes is a function displacements of the contacted nodes, namely $U_f = R(U_c)$. \tilde{R} is an approximation of R , based on the solution of $P(U) = \mathbf{f}$.

a convincing surgery simulation. Many elastic material models have been employed, for example, by Fung (1981), Meaney (2003), and Liu *et al.* (2004), in static analysis, some of which have been recently proposed for real time simulation (Picinbono *et al.* 2001, Wu *et al.* 2001 and Nienhuys 2001). We describe in the following subsections a formulation of the nonlinear finite element procedure, part of which may be subsequently employed for computing haptic feedback in real time.

2.1 Material constitutive models

A deformable model of soft tissue comprises a mathematical description of both the geometry and its

deforming mechanism, based on the composition and elasticity of its material components. While the fundamental theory of elasticity and its computational techniques are found in the literature (Ogden 1984, Zienkiewicz and Taylor 2000, Fu and Ogden 2001), here we introduce some basic concepts for completeness.

Suppose the three invariants of the symmetric right Cauchy-Green tensor are given by

$$I = C_{kk}, \quad II = (I^2 - C_{kl}C_{lk}), \quad III = \det C_{kl} = J^2,$$

then $J = 1$ represents a volume-preserving deformation. Without loss of generality, in this paper we consider the case of a compressible Neo-Hookean material model (Zienkiewicz and Taylor 2000) whose stored energy density is expressed as

$$W_{WF}(I, J) = \frac{\mu}{2}(I - 3 - 2 \ln J) + \frac{\lambda}{2}(J - 1)^2, \quad (1)$$

where the parameters μ and λ give the same response as Lamé parameters for small deformation (Zienkiewicz and Taylor 2000), and the term $2 \ln J$ was first introduced by Wall and Flory in 1950s (Peng and Chang 1997) for considering the effect of a change of volume. Its Cauchy stress can then be calculated as

$$\sigma_{ij} = \frac{\mu}{J}(g_{ij} - \delta_{ij}) + \lambda(J - 1)\delta_{ij}, \quad (2)$$

where g_{ij} are the left Cauchy strain tensors and δ_{ij} is the Kronecker delta function.

2.2 Formulation procedure of a non-linear finite element method

A set of nonlinear algebraic equations can be constructed by considering the variation of the general energy $\Pi := \int_{\Omega} W(\mathbf{E}) dV - \Pi_{\text{ext}}$ which is set to zero, i.e.

$$\int_{\Omega} \delta E_{IJ} S_{IJ} dV - \delta \Pi_{\text{ext}} = 0, \quad (3)$$

in the reference configuration. To discretize the problem domain, we adopt tetrahedral elements and use the natural shape functions $N_i(\zeta, \eta, \xi)$ as an interpolation basis. Since the external force \mathbf{f}_{ext} satisfies $\mathbf{f}_{\text{ext}} = \mathbf{f}_{\text{body}} + \int_{\gamma} N_i \mathbf{t} dv$, then under the current configuration, equation (3) yields

$$\int_{\omega} \mathbf{B}^T \boldsymbol{\sigma} dv - \mathbf{f}_{\text{ext}} = 0, \quad (4)$$

where \mathbf{B} is a constant matrix if $N_i(\zeta, \eta, \xi)$ is linear. The left Cauchy-Green strain g_{ij} in $\boldsymbol{\sigma}$ can be calculated from

$$\begin{aligned} g_{11} &= 1 + 2 \sum_{i=1}^4 b_i u_i + \left(\sum_{i=1}^4 b_i u_i \right)^2 + \left(\sum_{i=1}^4 c_i u_i \right)^2 \\ &\quad + \left(\sum_{i=1}^4 d_i u_i \right)^2 \\ g_{22} &= 1 + 2 \sum_{i=1}^4 c_i v_i + \left(\sum_{i=1}^4 b_i v_i \right)^2 \\ &\quad + \left(\sum_{i=1}^4 c_i v_i \right)^2 + \left(\sum_{i=1}^4 d_i v_i \right)^2 \\ g_{33} &= 1 + 2 \sum_{i=1}^4 d_i w_i + \left(\sum_{i=1}^4 b_i w_i \right)^2 \\ &\quad + \left(\sum_{i=1}^4 c_i w_i \right)^2 + \left(\sum_{i=1}^4 d_i w_i \right)^2 \\ g_{12} &= \sum_{i=1}^4 b_i v_i + \sum_{i=1}^4 c_i u_i + \left(\sum_{i=1}^4 b_i u_i \right) \left(\sum_{i=1}^4 b_i v_i \right) \\ &\quad + \left(\sum_{i=1}^4 c_i u_i \right) \left(\sum_{i=1}^4 c_i v_i \right) \\ &\quad + \left(\sum_{i=1}^4 d_i u_i \right) \left(\sum_{i=1}^4 d_i v_i \right) \\ g_{23} &= \sum_{i=1}^4 c_i w_i + \sum_{i=1}^4 d_i v_i + \left(\sum_{i=1}^4 b_i w_i \right) \left(\sum_{i=1}^4 b_i v_i \right) \\ &\quad + \left(\sum_{i=1}^4 c_i w_i \right) \left(\sum_{i=1}^4 c_i v_i \right) \\ &\quad + \left(\sum_{i=1}^4 d_i w_i \right) \left(\sum_{i=1}^4 d_i v_i \right) \\ g_{31} &= \sum_{i=1}^4 b_i w_i + \sum_{i=1}^4 d_i u_i + \left(\sum_{i=1}^4 b_i u_i \right) \left(\sum_{i=1}^4 b_i w_i \right) \\ &\quad + \left(\sum_{i=1}^4 c_i u_i \right) \left(\sum_{i=1}^4 c_i w_i \right) \\ &\quad + \left(\sum_{i=1}^4 d_i u_i \right) \left(\sum_{i=1}^4 d_i w_i \right) \end{aligned} \quad (5)$$

where b_i , c_i , d_i are the partial derivatives of N_i . After integration of equation (4) over all the elements, we obtain a set of non-linear algebraic equations

$$\boldsymbol{\Phi}(\mathbf{U}) = \mathbf{f}_{\text{ext}}(\mathbf{U}) - \mathbf{P}(\mathbf{U}) = \mathbf{0}, \quad (6)$$

where $\mathbf{P} = (P_1, \dots, P_{3n})^T$ is a vector-value function, with its components P_i being computed through $P_i(\mathbf{U}) = \sum_{j \in Q_i} \int_{\omega_j} \mathbf{B}^T \boldsymbol{\sigma} dv$, where Q_i are the neighbors of the node p_i . If we only consider the steady-state problem in which the initial stress and initial strain are ignored, then

the force term can be generated from

$$\mathbf{f}_{\text{ext}}^{(j)} = \sum_{\alpha=1,2,3,4} \left(\int_{\omega_j} N_{\alpha} \rho_0 \mathbf{b}^{(m)} dv \right) + \sum_{\beta=1,2,3} \left(\int_{\gamma_j} N_{\beta} \bar{\mathbf{t}} dS \right) \quad (7)$$

where $\mathbf{b}^{(m)}$ and $\bar{\mathbf{t}}$ are the vectors of the body forces and the averaged traction forces, respectively. To evaluate the integral $\int_{\gamma_j} N_{\beta} dS$ on a surface triangle face γ_j of a tetrahedron, we translate it to its local coordinates

$$\int_{\gamma_j(x,y,z)} N_{\beta} dS = \iint_{\gamma_j(\zeta,\eta,\xi)} N_{\beta} D_j d\zeta d\eta \quad (8)$$

where

$$D_j = \begin{bmatrix} \frac{\partial x}{\partial \zeta} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \xi} \end{bmatrix} \otimes \begin{bmatrix} \frac{\partial y}{\partial \zeta} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \xi} \end{bmatrix}.$$

Consequently, when the surface face is assumed to be $\xi = 0$ in the un-deformed state, N_{β} can be simplified to $\zeta, \eta, 1 - \zeta - \eta$. The force on each vertex of the triangle may be generated from

$$\iint_{\gamma_j(\zeta,\eta,\xi)} N_{\beta} D_j d\zeta d\eta = D_j \int_0^1 d\zeta \int_0^{\zeta} N_{\beta} d\eta = \frac{D_j}{6}, \quad (9)$$

$$\beta = 1, 2, 3.$$

The above procedure establishes a non-linear algebraic system. In the following sections, we focus on solving this system of equations.

2.3 Numerical solution to non-linear algebraic equations

As shown above, $\Phi(U)$ is a multi-dimensional non-linear function, and the Jacobean matrix $\left(\frac{\partial \Phi}{\partial U}\right)$ of $\Phi(U)$ can be specified as

$$\left(\frac{\partial \Phi}{\partial U}\right) = \left(\int_{\omega} B_{\alpha}^T D^c B_{\beta} dv + \int_{\omega} \frac{\partial N_{\alpha}}{\partial \zeta} \sigma_{\zeta\eta} \frac{\partial N_{\beta}}{\partial \eta} dv \right), \quad (10)$$

For the Neo-Hookean model, the element of D^c under the current configuration is $d_{ijkl} = \lambda(2J - 1)\delta_{ij}\delta_{kl} + ((\mu/J) - \lambda(J - 1))(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk})$. The derivative information is therefore calculated explicitly, and the nonlinear algebraic system (6) is consequently solved using the Newton–Raphson method, which converges rapidly to the root of $\Phi(U)$. However, as a non-linear vector-valued function, it may have multiple solutions. To ensure that the computed result is the solution of such equations, we begin from a known solution (in our case, we assume $\Phi(0) = 0$), and then gradually update the external force \mathbf{f}_i . With Newton's method, we calculate the Jacobean matrix $\left(\frac{\partial \Phi}{\partial U}\right)$ of $\Phi(U)$ at the current location to obtain the direction of its tangent, and

then find the next location through the intersection of \mathbf{f}_i with the tangent line (figure 1b). We iterate this process until we find the point where the final \mathbf{f}_i ($= \mathbf{f}$) meets \mathbf{P} .

From the procedure used in the above construction, we know that each component P_i contains only the displacement variables of its corresponding neighbors, so the Jacobean matrix is sparse. Using this property, the value of the next iterative step can be found by the conjugate gradient method (which has a computational cost of $O(n)$ compared to $O(n^3)$ for LU decomposition).

Even though for each \mathbf{f}_i , the Newton–Raphson method converges rapidly, it needs two additional procedures: one to control the stepsize of all the displacement variables so that the next step will stay in the attractive zone, and the other to determine the growth and sign of \mathbf{f}_i , so that the correct solution can be tracked.

As shown in Zienkiewicz and Taylor (2000), a line search method can solve the first problem, but it needs to evaluate the function $\Phi(U)$ several times, which is time consuming. Here we use a backtracking method that is often the best means of determining stepsize (Press *et al.* 1988). For the second problem we use Bergan's procedure (Zienkiewicz and Taylor 2000) to control the updating behavior of \mathbf{f} . With the two additional loops to guarantee the convergence to the correct solution, the speed of the non-linear FE procedure is at least ten times slower than the speed of computing a linear system using the conjugate gradient approach only once. A computed result on a standard cylinder model using the above procedure is shown in figure 2b.

For non-linear and non-quadratic functions, the rapid convergence of the Newton–Raphson method may be hindered by roundoff errors, scaling problems, or inappropriately chosen parameter values. Dealing with such problems in real time is tedious, so we address these issues at the preprocessing stage, while at the time of actual simulation, we directly use the DD relations as shown in figure 1c.

2.4 Validation of the nonlinear FE solver

In our implementation, we use an input file format similar to that employed by the commercial software ABAQUS® (ABAQUS, Inc., Providence, RI). To test the above procedure, we used ABAQUS to create a basic cylinder model 30 cm in radius and 20 cm high, which is meshed into 195 nodes and 586 tetrahedra. We set the material parameters in the Neo-Hookean model as $\mu = 0.48$ and $\lambda = 0.2$, which results in a fairly compressible model according to the standard specified in ABAQUS. The base of our model is fixed on the xy -plane and centered at the origin. The external force is exerted at the center of the top face at $P_1(0,0,20)$. Based on this configuration, it is clear that the nodes with the maximal displacements will occur on the top of the cylinder.

Taking the results from ABAQUS as a gold standard, we check the displacements of P_1 and the other two extreme points $P_2(0,30,20)$ and $P_3(30,0,20)$ as shown in figure 2b.

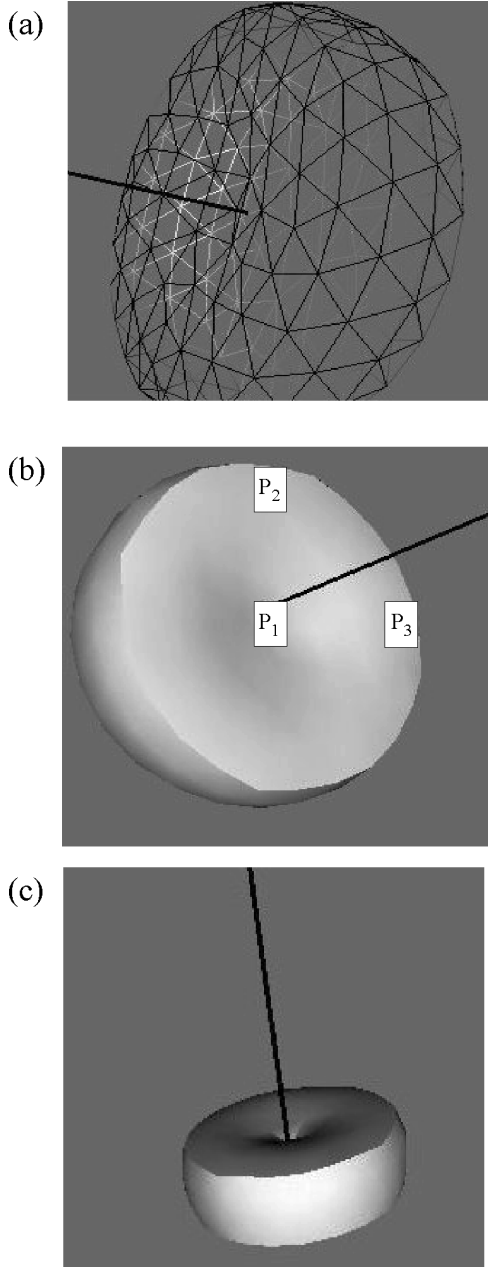


Figure 2. (a) A mesh of a cylinder with 195 nodes and 586 tetrahedra. (b) A needle probing at the center P_1 with the developed FE solver; The nodes $P_2(0,30,20)$ and $P_3(30,0,20)$ are used for comparison with ABAQUS. (c) A large deformation using the interpolation of the DD-function generated from the nonlinear FE solver.

The dominant displacements of P_1 , P_2 and P_3 are in the z , y , x directions, respectively. Since the model is fixed over its entire base, there is no obvious rotational behavior. Thus the results from ABAQUS and our “Robarts” solver (Rob) show that the z -displacement of P_1 demonstrates a strong linearity with the external force (figure 3a,b).

For free nodes, the dominant displacements of P_2 and P_3 are also linear with respect to the external force, as well as to the z -displacement of P_1 (figure 4a,b).

As an example of the DD relation, the y and z displacements of P_3 generated by ABAQUS are characterized in terms of the z -displacement of P_1 (figure 5a).

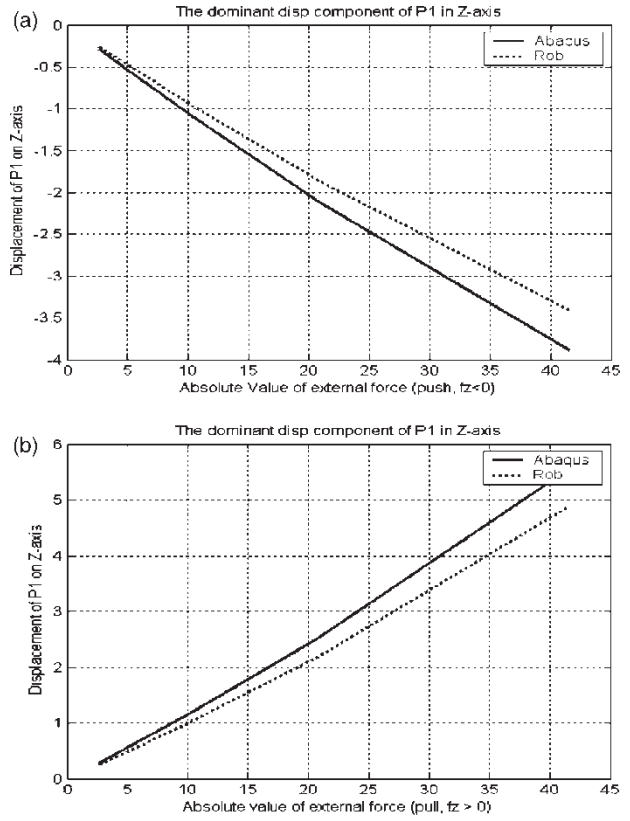


Figure 3. (a) The dominant displacement of P_1 vs. the external probing force. (b) The dominant displacement of P_1 vs. the external pulling force.

Note that there are some constant deviations (around 11%) between the results, which are due to the kinematic constraint condition of incompressibility ($J = 1$). ABAQUS uses the modified deformation gradient $\tilde{F} = J^{-1/3}F$ to accommodate the compressive effect (Ogden 1984). Its strain energy density function $W_{\text{Aba}}(I, J)$ is defined as

$$W_{\text{Aba}}(I, J) = \frac{\mu}{2}(J^{-2/3}I - 3) + \frac{\lambda}{2}(J - 1)^2, \quad (11)$$

while $W_{\text{WF}}(I, J)$ defined in equation (1) is Wall and Flory’s formula, which includes the term “ $\ln J$ ” derived from a penalty function to ensure that the stresses are zero in the unstrained state (Peng and Chang 1997, Holzapfel and Ogden 2003). Peng and Chan showed that the introduction of the penalty function is equivalent to the modification of the strain energy function from an incompressible to a slightly compressible one. A thorough discussion of this aspect can be found in Peng and Chang (1997).

In fact, if we compare the shear terms in equations (1) and (11), and set

$$\begin{aligned} \Phi(I, J) &= W_{\text{Aba}}(I, J) - W_{\text{WF}}(I, J) \\ &= \frac{\mu}{2}(J^{-2/3}I - I + 2 \ln J), \end{aligned} \quad (12)$$

it is easy to verify that $(\partial\Phi/\partial J) = \mu J^{-5/3}(-I/3 + J^{2/3}) < 0$. From $\Phi(I, 1) = 0$, it can be shown that the energy variation generated from $W_{\text{Aba}}(I, J)$ is greater than that obtained with Wall and Flory’s formula. As an

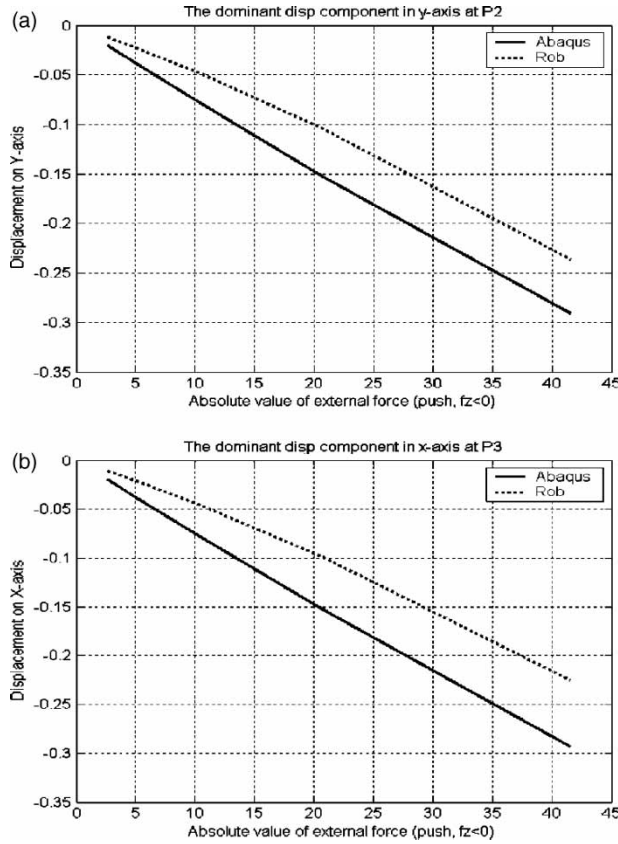


Figure 4. (a) The dominant component results of P_2 on the Y -axis vs. the external pushing force. (b) The dominant component results of P_3 on the X -axis vs. the external pushing force.

example, the shear terms of W_{Aba} and W_{WF} for the case of the principle invariant $I = 4.2$ are plotted in figure 5b. Since under a given stress, the strain energy density is proportional to the strain as well as the displacement, ABAQUS consequently yields a larger displacement than our method.

To demonstrate that the nonlinear system is sensitive to this shear term, we replace $(I - 3 - 2 \ln J)$ in equation (1) by $(I - 3 - 0.998 * 2 \ln J)$. Then the difference in the dominant displacement components as shown in figure 3a can be reduced from 11.6 to 1%. However, with this substitution the stresses in the unstrained state are no longer zero, so the material model is corrupted. But since soft tissue is incompressible or nearly so, the ratio λ/μ usually varies from orders of hundreds to thousands (ABAQUS, Usyk and McCulloch 2003, Chabanas *et al.* 2004). When the previously used ratio of $\lambda/\mu = 0.416$ is increased by a factor of 100 and 1000, the displacement error is reduced from 11.6 to 3.4 and 0.7%, respectively.

In this experiment, the relationship between displacement and force exhibits strong linearity even with deformations of up to 18% of the thickness of the cylinder. This linearity is due to the fact that we fixed the entire base of the model so that there was no obvious rotation. However, we can still see that there is a nonlinear property in some non-dominant components, as shown in figure 5a.

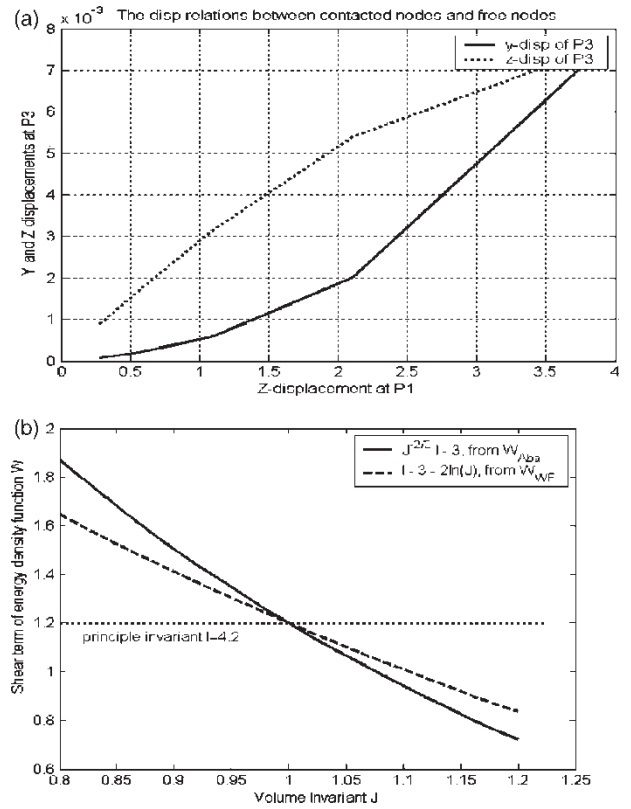


Figure 5. (a) The non-dominant displacements of the free node P_3 vs. the z -displacement of P_1 . (b) Neo-Hookean model formations used by the ABAQUS and our implementation.

With a small mesh size and small forces, the full FE solver can perhaps achieve real time performance, but it certainly cannot meet the requirements of a real time simulation in the general case. In the following sections, we discuss some real time techniques that are applicable to the simulation of tissues exhibiting both linear and nonlinear behaviors.

3. Real time approaches

In the previous section, we established a set of nonlinear algebraic equations. In a surgical simulation, a probe or surgical tool contacts only a few elements of the mesh. Suppose $S = \{s_1, s_2, \dots, s_k\}$ is a set of the contacted nodes, and $\{u_i, v_i, w_i, i = 1, \dots, k\}$ is the set of their displacements, where k is a small number. Suppose $d_{\{i,S\}}^{(u)}$ denotes the set of the u -displacements for all the neighbors of the free node p_i , then equation (6) can be reduced to implicit functions:

$$\begin{cases} P_{3i}(d_{\{i,S\}}^{(u)}, u_1, v_1, w_1, \dots, u_k, v_k, w_k) - f_{3i}^u = 0 \\ P_{3i+1}(d_{\{i,S\}}^{(v)}, u_1, v_1, w_1, \dots, u_k, v_k, w_k) - f_{3i+1}^v = 0 \\ P_{3i+2}(d_{\{i,S\}}^{(w)}, u_1, v_1, w_1, \dots, u_k, v_k, w_k) - f_{3i+2}^w = 0 \end{cases}$$

$$i = 1, \dots, n - k.$$

(13)

In practice, each displacement $\{d_i^{(u)}, d_i^{(v)}, d_i^{(w)}\}$, $i = 1, \dots, n - k$, is determined by the displacements $\{u_i, v_i, w_i\}$, $i = 1, \dots, k$, enabling equation (13) to be rewritten as

$$\begin{cases} d_i^{(u)} = R_{3i}(u_1, v_1, w_1, \dots, u_k, v_k, w_k) \\ d_i^{(v)} = R_{3i+1}(u_1, v_1, w_1, \dots, u_k, v_k, w_k), \\ d_i^{(w)} = R_{3i+2}(u_1, v_1, w_1, \dots, u_k, v_k, w_k) \end{cases} \quad (14)$$

$i = 1, \dots, n - k,$

where R_i are the functions of the displacements of the contacted nodes (called the DD-functions). While we do not give a mathematical proof of the existence of the implicit function R_i , we can show below that equation (14) is an explicit linear expression for a linear model, and for nonlinear elastic models, R_i in equation (14) is generally nonlinear and only can be obtained through approximation approaches. The explicit DD relation described below can achieve real time performance and is more flexible than the pre-computed approach in Cotin *et al.* (1999).

3.1 Explicit displacement relations in linear elastic models

In the FEM, the classic linear model generates the following matrix equation for each element e_i : $K^{e_i} D^{e_i} = f^{e_i}$, where K^{e_i} is the stiffness matrix (12×12 for a tetrahedron), and D^{e_i} and f^{e_i} are the vectors of the displacements and forces of the element's nodes. After assembling these individual forces at each node, the global equation can be written as

$$K(d_1 d_2 \dots d_{3n})^T = (f_1 f_2 \dots f_{3n})^T, \quad (15)$$

where the f_i are external forces, and K is the assembled global stiffness matrix. Similar to the condensation technique used in Bro-Nielsen and Cotin (1996) which separates all the nodes into the internal and surface components to reduce the size of K for efficiency, we divide all the nodes into three sets: the contacted nodes, the free nodes and the fixed boundary nodes, denoted by $P_K = (p_1 p_2 \dots p_k)$, $P_R = (p_{k+1} \dots p_m)$ and $P_B = (p_{m+1} \dots p_n)$, respectively. The above global equations are then reduced to:

$$\tilde{D} = \tilde{K}^{-1} \tilde{f}, \quad (16)$$

where \tilde{K} is a $3m \times 3m$ matrix obtained from K by removing the rows and columns corresponding to the set of the fixed nodes. \tilde{D} and \tilde{f} correspond to all other nodes except those that are fixed. Without loss of generality, we may write \tilde{K}^{-1} as

$$\tilde{K}^{-1} = \begin{pmatrix} A_{3k \times 3k} & B_{3k \times 3(m-k)} \\ C_{3(m-k) \times 3k} & D_{3(m-k) \times 3(m-k)} \end{pmatrix}, \quad (17)$$

where the $A_{3k \times 3k}$ corresponds to the contacted nodes P_K . To simplify the following expression, we ignore the effect of gravity so that the external forces at the free nodes P_R

are zero. The above equations yield

$$(f_1 f_2 \dots f_{3k})^T = A_{3k \times 3k}^{-1} (d_1 d_2 \dots d_{3k})^T. \quad (18)$$

Combining equations (16) and (18), we obtain:

$$\begin{aligned} \begin{pmatrix} d_1 \\ \dots \\ d_{3m} \end{pmatrix} &= \tilde{K}_{3m \times 3k}^{-1} A_{3k \times 3k}^{-1} \begin{pmatrix} d_1 \\ \dots \\ d_{3k} \end{pmatrix} \\ &= \begin{pmatrix} I_{3k \times 3k} \\ C_{3(m-k) \times 3k} A_{3k \times 3k}^{-1} \end{pmatrix} \begin{pmatrix} d_1 \\ \dots \\ d_{3k} \end{pmatrix}, \end{aligned} \quad (19)$$

where $\tilde{K}_{3m \times 3k}^{-1}$ represents the $3k$ columns of \tilde{K}^{-1} corresponding to the set P_K .

In contrast to Cotin *et al.* (1999) where the FEM solver needs to run $3k$ times to generate a tensor matrix for each node, the above method generates each tensor matrix directly from the inverse of the global stiffness matrix $\tilde{K}_{3m \times 3m}$. So when the set of the contacted nodes K is replaced by another set K_I , we only need to renumber A , B , C , D in $\tilde{K}_{3m \times 3m}^{-1}$ and then recalculate $A_{3k \times 3k}^{-1}$.

In addition to allowing the contacted nodes to be displaced in real time, the above approach also allows users to dynamically add boundary conditions during the real time simulation. We may first fix l ($l \geq 3$) nodes to generate the global inverse $\tilde{K}_{3(n-l) \times 3(n-l)}^{-1}$. During the simulation time, if an additional node p_j ($p_j \in P_R$) is to be fixed by the user, we may extend $A_{3k \times 3k}$ to $A_{3(k+1) \times 3(k+1)}$ to include three extra rows and columns for p_j . Equation (18) is replaced by

$$\begin{aligned} &(f_1, \dots, f_{3k}, f_j, f_{j+1}, f_{j+2})^T \\ &= A_{3(k+1) \times 3(k+1)}^{-1} (d_1, \dots, d_{3k}, d_j, d_{j+1}, d_{j+2})^T \\ &= A_{3(k+1) \times 3k}^{-1} (d_1, \dots, d_{3k})^T. \end{aligned} \quad (20)$$

Therefore, the analytic linear relationship between displacements of the contacted nodes and free nodes can be specified as:

$$(d_1 d_2 \dots d_{3m})^T = \tilde{K}_{3m \times 3(k+1)}^{-1} A_{3(k+1) \times 3k}^{-1} (d_1 d_2 \dots d_{3k})^T. \quad (21)$$

Since k is the number of the contacted nodes (which usually is small, e.g. $k < 3$), the concise tensor matrix $T = \tilde{K}_{3m \times 3k}^{-1} A_{3k \times 3k}^{-1}$ gives a greatly improved speed compared to the linear solver. In fact, the solution of equation (19) may involve $(3k)^3$ multiplications to compute the inverse of A and $(9k^2)(3m)$ multiplications for the expression $\tilde{K}_{3m \times 3k}^{-1} A_{3k \times 3k}^{-1}$, or a total of $27k^2(m+k)$ operations. If we suppose that the linear solver uses the conjugate gradient method, it needs at least $60m\nu$ multiplications (see Zhong *et al.* (2005)) where the loop number ν is about 100. So if k is less than 3, the pre-computed approach is at least 10 times faster than the CG

method, but with the same accuracy. Results obtained using these approaches are given in Section 3.4. The time required to formulate the global stiffness matrix may vary, depending on its implementation as well as the model's configuration. With our current implementation on a 1 GHz Linux computer, the time for building a $11,763 \times 11,763$ matrix is 4.69 s. Fortunately for the probing simulation, the matrix does not require updating in either the pre-computed procedure or in the CG method, except when its configuration changes.

We demonstrate below that such an explicit linear relationship cannot be obtained for non-linear elastic material.

3.2 Polynomial interpolation to non-linear FE model

In the previous section, we demonstrated that the tensor matrix method is valid because the relationship between the force and displacement on each element is linear, and this linearity propagates in FE methods. But for hyper-elastic materials, the displacements of each node is no longer linear with respect to the applied forces (see figure 1a and 5a), so compared to the linear elastic model where $\{d_i^{(u)}, d_i^{(v)}, d_i^{(w)}\}$ is expressed exactly in linear terms of $\{u_i, v_i, w_i\}$ as in equation (14), hyper-elastic material models only can generate implicit expressions (13). For such a $\{d_i^{(u)}, d_i^{(v)}, d_i^{(w)}\}$, there is usually no analytic expression like equation (19). However, we can approximate the functions R_ν in equation (14) with a polynomial \tilde{R}_ν . For the convenience of discussion, we assume S contains only one node. To approximate the function R_ν , we choose a basis $\{1, X_j\}$ so that each displacement d_ν can be expressed as

$$d_\nu \approx \tilde{R}_\nu(X_j) = a_{0,\nu} + \sum a_{j,\nu} X_j. \quad (22)$$

Suppose that the basis $\{1, X_j\}$ is equal to $\{1, u, v, w, uv, vw, wv, uv, vw, uv, vw, wu, u^2, v^2, w^2\}$. With the above FEM solver, we generate N^3 sets of displacements for each free node based on the N^3 given sets $\{u^{(i)}, v^{(j)}, w^{(k)}\}$, $j, k = 1, \dots, N$ of displacements of the contacted node (or the needle tip).

To calculate the coefficients $a_{j,\nu}$ for a better approximation, we assume the maximal displacement of the contacted points before the surface ruptures to be $\mathbf{r}_M = \mathbf{r}_M(u_M, v_M, w_M)$. Partitioning $(0, \mathbf{r}_M)$ into N^3 ($N^3 \geq 9k + 1$) sub-domains and taking each

$$\mathbf{r}_M^{(i,j,k)} = \left(\frac{u_M}{N} i, \frac{v_M}{N} j, \frac{w_M}{N} k \right)$$

as a prescribed constraint on the contacted node in the FE model, then we use the FE solver to calculate the displacements $\{x_p^{(i)}, y_p^{(j)}, z_p^{(k)}\}$ of each free node p . Consequently, the coefficients $a_{i,p}$ for the x-displacement

of the node p satisfies

$$\begin{aligned} \mathbf{Q} \begin{pmatrix} a_{0,p} \\ a_{1,p} \\ \dots \\ a_{9,p} \end{pmatrix} &:= \begin{pmatrix} 1 & X_1^{(1)} & X_2^{(1)} \dots X_9^{(1)} \\ 1 & X_1^{(2)} & X_2^{(2)} \dots X_9^{(2)} \\ \dots & \dots & \dots \\ 1 & X_1^{(N^3)} & X_2^{(N^3)} \dots X_9^{(N^3)} \end{pmatrix} \begin{pmatrix} a_{0,p} \\ a_{1,p} \\ \dots \\ a_{9,p} \end{pmatrix} \\ &= \begin{pmatrix} x_p^{(1)} \\ x_p^{(2)} \\ \dots \\ x_p^{(N^3)} \end{pmatrix}. \end{aligned} \quad (23)$$

Here we may simply invert \mathbf{Q} to obtain $a_{i,p}$ if N^3 equals to the number of the interpolation basis functions in equation (22). Otherwise, equation (23) may be solved using the least squares method, i.e. we have

$$(a_{0,p} \ a_{1,p} \dots a_{9,p})^T = (\mathbf{Q}^T \mathbf{Q})^{-1} \mathbf{Q}^T (x_p^{(1)} \ x_p^{(2)} \dots x_p^{(N^3)})^T. \quad (24)$$

With the computed coefficients $a_{i,p}$ for each p , the simulation of real time deformation only involves the multiplication of the coefficient vector with the displacements of contacted nodes. With the obtained displacements, we can further calculate the force or pressure feedback locally through equations (4) and (7).

The basis $\{1, x_j\}$ used above is quadratic but it could be linear, cubic or higher order depending on the accuracy required. Even in the linear form, its approximation is still different from the tensor matrix (Cotin *et al.* 1999) because its original DD-functions are nonlinear.

3.3 St. Venant's principle and affine mapping

In real time simulation, surgeons generally require interaction with the model in arbitrary regions, and the approach described in the previous section is limited to the constraints defined in the preprocessing stage. To overcome this problem, we employ St. Venant's principle in elastic theory (Eisley 1989) for an approximate result. St. Venant's principle states that statically equivalent systems of forces produce the same stresses and strains within a body except in the immediate region where the loads are applied. It suggests that the stress change in small vicinity is virtually independent of other parts of the body. Now suppose $T = \{t_1, t_2, \dots, t_k\}$ is a set of nodes contacted by the needle during a dynamic deformation, located physically close to the set S discussed previously. We separate the region V containing both S and T . Suppose b_1 and b_2 are two nodes on the boundary ∂V of V , and $s = (x_s, y_s, z_s)$ and $t = (x_t, y_t, z_t)$ are the centers of S and T . According to St. Venant's Principle, we construct an affine mapping ϕ of V so that the displacement and force feedback in V can be adjusted

according to different probe regions, but outside of V , they remain unchanged. To construct such a mapping, let ϕ satisfy $\phi(b_1) = b_1$, $\phi(b_2) = b_2$, $\phi(s) = t$, generating the following relation:

$$\begin{aligned} \begin{pmatrix} x_{b_1} & x_{b_2} & x_t \\ y_{b_1} & y_{b_2} & y_t \\ z_{b_1} & z_{b_2} & z_t \end{pmatrix} &= \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x_{b_1} & x_{b_2} & x_s \\ y_{b_1} & y_{b_2} & y_s \\ z_{b_1} & z_{b_2} & z_s \end{pmatrix} \\ &= \phi \begin{pmatrix} x_{b_1} & x_{b_2} & x_s \\ y_{b_1} & y_{b_2} & y_s \\ z_{b_1} & z_{b_2} & z_s \end{pmatrix}. \end{aligned} \quad (25)$$

where ϕ may be solved by inverting the 3×3 matrix on the right hand side of equation (25). Now ϕ establishes a relationship between the two coordinates, i.e. if p is a free node in V , its coordinates are mapped to $q'(x', y', z')$ under ϕ . Since each polynomial in the previous approximation is associated with a node, we need to find the node q closest to q' , so that we may use the pre-computed polynomial \tilde{R}_q for p .

To avoid searching the entire index table for q , we establish a local table for each node. The size of the table depends on the distance between S and T . If the distance is increased, V is expanded until ∂V meets P_B and then b_1 and b_2 may be chosen from P_B . But this is the worst case for ϕ which is designed based on the St. Venant's principle for a small region V . The effect of such a mapping (i.e. $b_1, b_2 \in P_B$) is shown in figure 6c.

In this work, we use two sets of mesh data, one the cylinder, consisting of 586 tetrahedra generated from ABAQUS CAE, the other, a cortical brain phantom whose mesh was generated from MRI data with the marching cubes algorithm (Lorenson and Cline 1987) and regularized by the ICEM® (ICEM CFD Engineering, ANSYS, Inc., Berkely, CA) program. The final mesh includes 3921 nodes and 11,741 tetrahedra.

3.4 Results on real time simulation approach

In the previous sections we introduced two simulation approaches: the enhanced pre-computed FE method (EPFE) whose computation formulae are specified in equations (19) and (21) for the linear elastic model, and the polynomial interpolation with an affine mapping (PIAM) for non-linear elastic models. Both EPFE and PIAM were tested by simulating the probing of the brain model. Compared to the pre-computed FE model (PFE) in Cotin *et al.* (1999) which involves approximately $3m \times 3k$ multiplications for each operation during the simulation, the proposed EPFE method requires $3m \times (3k + 3k) + k^3$ multiplications, where m and k denote the numbers of free nodes and the nodes in the contacted area respectively, and where k^3 operations are required to invert the small sub-matrix. If $k \ll m$ (e.g. $k = 3$, $m = 1000$), the time to convert the sub-matrix is small relative to the overall computation time. Generally the three pre-computed FE methods PFE, EPFE and PIAM are potentially

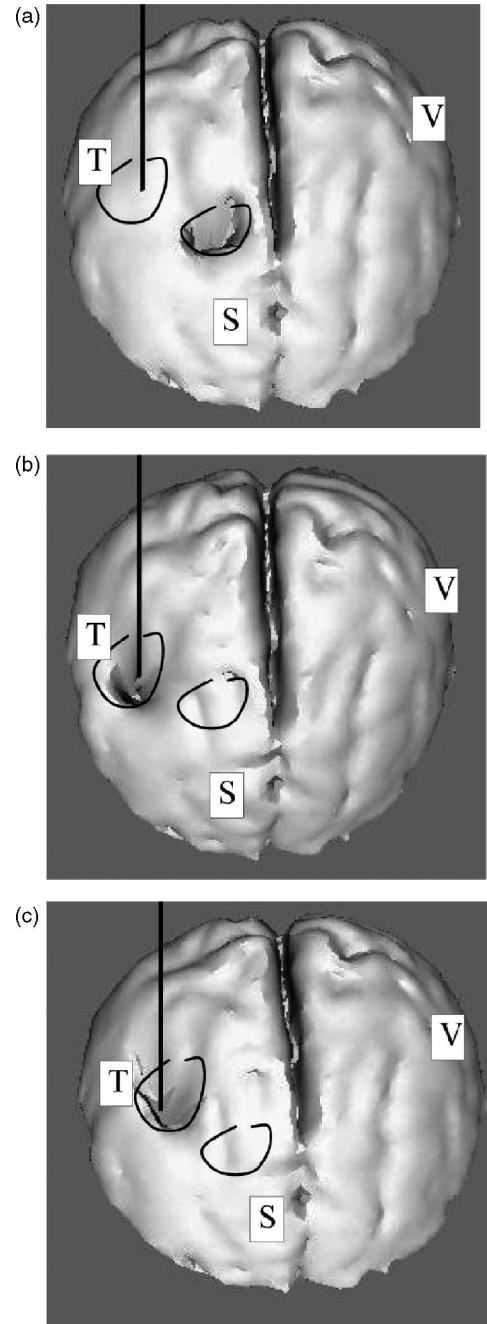


Figure 6. S and T denote the probe/surface-contacted areas in the preprocessing stage and a real time animation, respectively. Probing results shown above are obtained through the following methods: (a) PFE: the probe at T but the deformation centered on S . (b) EPFE: the deformation is centered on T as expected. (c) PFE with Affine Mapping: it transfers the pre-computed deformation centered on S to that on T .

at least 10 times faster than the localized conjugate gradient method (LCG) described in Nienhuys and van der Stappen (2000). The four methods were implemented using the tetrahedral mesh of the brain phantom containing 2886 free nodes, generated with ICEM. Results of this study are presented in table 1.

Although PFE computes deformations rapidly, its preprocessing step needs to be repeated to re-compute the linear system when the interaction occurs outside the

Table 1. Four real-time finite element approaches are compared.

	LCG	PFE	EPFE	PIAM
Frames/s	3	200 +	50–100	30–40
Flexibility	Very good	Poor	Good	Good

pre-defined area; otherwise, it becomes inaccurate, clearly demonstrated by figure 6a. On the other hand, the deformation simulated by EPFE can be driven by arbitrary points contacted by the probe without repeating the preprocessing step, so it is more flexible than PFE (figure 6b). An affine mapping can correct the error from PFE (figure 6c), and is also applicable to non-linear elastic models.

When gravity is ignored, EPFE and a linear solver are only different in their manipulation of the same linear algebraic equations. They exhibit the same accuracy except for some round-off errors (less than 10^{-4}), whether a linear solver directly uses the inverted global stiffness matrix or is solved via the LCG method. Since PFE is actually a linear interpolation of the solution from a linear solver, its tensor matrix is exactly the expression (19) of EPFE when the body force is ignored. So in the linear case, both PFE and EPFE have the same accuracy as that of an exact solver (such as LCG), with the main difference being in their flexibility.

Next we test the accuracy of the interpolation for the nonlinear model. For the convenience of implementation, we generate the displacements $\mathbf{r}_M^{(i,i)}$ of the probe tip P_1 by $F_M^i = (0, 0, -8i)$, $i = 1, \dots, 10$. Without loss of generality, the validation focuses on the given force direction. For example, the polynomial for the z -displacement of a node $P_m(-9.42, 5.32, 20)$ is computed as

$$\begin{aligned} \tilde{R}_{m,z} = & 0.0000006 + 5.05423u + 1.03084v \\ & + 0.0989905w + 4.28168uv - 0.458528vw \\ & + 1.53791wu - 19.7585u^2 \\ & - 3.22194v^2 - 0.0289002w^2. \end{aligned}$$

Using this polynomial we test nine data sets generated by $\tilde{F}_M^i = (0, 0, -8i + 4)$, $i = 0, \dots, 8$.

The results in table 2 show that the maximal error increases with \tilde{F}_M^i and could be up to 43% when the

maximal strain is about 23%. While the error is contributed partly from the second order approximation, we will see that the main part of it arises from the sampling data set itself.

For the sake of convenience, we accumulate \tilde{F}_M^i each time in the generation of the overall coefficient matrix, while in the validation procedure, we directly specify these forces in the input files. Since the relation between force and displacement is not linear, the displacement generated from the sum of the force is not equal to the sum of their displacements, as demonstrated in table 3.

In practice, the actual error from the interpolation should be much smaller than that shown in table 2. Also precision could be improved if the variables are sampled in all the three directions. The deformation from the simulation using the interpolation approach is shown in figure 2c.

To evaluate the effect of the affine mapping, we also compared the PFE approaches with or without an affine mapping. The experimental results show that the mean error of the displacements in the contacted nodes in the preprocessing step and the nodes in the real-time simulation was reduced from 5.9 mm (only PFE) to 0.63 mm (PFE with affine mapping).

4. Discussion

We have described the simulation of soft-tissue deformation using the DD-functions. In linear models, these functions are also linear. The tensor matrix approach used in Cotin *et al.* (1999) is actually a linear interpolation of the DD-functions, so several sets of the sampling data must be generated at significant time cost. Consequently, the boundary constraint and contacted locations in a real time animation are restricted to be the same as those employed in the preprocessing stage. Here, we directly construct the DD-functions through the inverse of the global stiffness matrix without using the sampling procedure, thus removing the restriction on the contacted locations as well as the boundary conditions. This modification provides the advantages of probing different regions, and fixing more boundary nodes on the fly.

However, accuracy is the most important feature in a simulation system. The classic linear model not only fails

Table 2. Nine sets of boundary constraints at P_1 are generated through nine sets of given forces.

\tilde{F}_M^i	$\mathbf{r}_M^{(i,i)}$ at P_1	z -displacement of P_m by FE solver	$\tilde{R}_{m,z}(\mathbf{r}_M^{(i,i)})$	%
(0, 0, -4)	(-0.00647673, 0.0231327, -0.327541)	-0.0413444	-0.0409	-1.075
(0, 0, -12)	(-0.0212289, 0.069491, -0.955437)	-0.123739	-0.1258	1.6656
(0, 0, -20)	(-0.0388128, 0.115715, -1.55091)	-0.206285	-0.2172	5.2912
(0, 0, -28)	(-0.0590201, 0.160393, -2.11458)	-0.288477	-0.3163	9.6448
(0, 0, -36)	(-0.0817335, 0.202718, -2.64742)	-0.37078	-0.4252	14.6772
(0, 0, -44)	(-0.106694, 0.241804, -3.14994)	-0.453104	-0.5462	20.5463
(0, 0, -52)	(-0.133486, 0.276377, -3.62138)	-0.535066	-0.6810	27.274
(0, 0, -60)	(-0.161659, 0.304977, -4.05941)	-0.616462	-0.8310	34.8015
(0, 0, -68)	(-0.190641, 0.32626, -4.46167)	-0.696764	-0.9960	42.9465

The results from the FE solver and the polynomial interpolation are compared.

Table 3. Two different displacements are generated: one in validation and other in the generation of the coefficients.

\bar{F}_M^i	$\mathbf{r}_M^{(i,i)}$ at P_1	z-displacement of P_m by FE solver	$\bar{R}_{m,z}(\mathbf{r}_M^{(i,i)})$	%
(0,0, - 80)	(-0.234457, 0.34139, -4.99196)	-0.813513	-1.2704	56.1622
Accumulated over F(0,0, - 8) ten times	(-0.171458, 0.438812, -6.1936)	-0.7800	-0.7800	0

to reflect the typical nonlinearity of the response of soft tissue, but it is also not invariant under rotation. This paper addressed the first problem and implemented the Neo-Hookean elastic model. With this model, we may simply compare the behavior of linear and non-linear models, since its two parameters μ and λ give the same response in the case of small deformations as for the linear elastic material with Lamé parameters. Also, the compressible behavior of the model can be easily interpreted through the ratio λ/μ .

Nevertheless, the nonlinear FEM has disadvantages for real time simulation. Firstly, it requires multi-layer loops to solve the system and each loop involves the computationally intensive evaluation of criteria functions, so it is time consuming. Secondly, the nonlinear solver is often divergent due to mesh quality, step size, initial force, scaling or round-off errors, and these problems cannot be solved easily in real time. To address these issues, we changed the problem of solving a nonlinear system to that of finding an approximation of the DD-functions in a preprocessing step, and we use the derived polynomials to achieve real time performance. Theoretically, this approximation may achieve any order of accuracy and its real time simulation performance is comparable to that of the linear case.

Unfortunately, it is difficult to make the preprocessing approach for nonlinear FEMs as flexible as for the linear FE model. To address this limitation, we proposed the use of the affine mapping approach based on the St.Venant principle for small variations in the contacted location on the surface of the model. This method provides an approximate means of relocating the contacted nodes for non-linear models. If the distance between S and T is small, the mapping ϕ can provide an appropriate displacement and haptic feedback for different probing positions in V , and St.Venant's principle permits the regions outside of V to retain the same response as that computed in the preprocessing stage. But if the distance is increased, the distribution of the displacements in V , especially in the contact area, will be reshaped due to the property of the affine mapping (see figure 6c).

Generally, a preprocessed FE approach has advantages in speed and accuracy, but is limited to some predefined constraints, especially for nonlinear models. For example, the preprocessed linear FE model could be adapted to an incision problem (Zhong *et al.* 2005) while it is much more difficult to do so using a nonlinear FE approach. Consequently, there is a tradeoff between accuracy and flexibility for linear or nonlinear FE models. If, in a real time animation, the rotation behavior is negligible and

deformation is small, a preprocessed linear finite element model is an appropriate choice.

5. Conclusion

In this paper, we have proposed the DD-functions to simulate the probing a soft-tissue mass. In particular, we directly constructed the linear DD-functions and demonstrated that this approach had advantages not present in the previously reported modeling techniques. Since soft tissue is inherently non-linear in its behavior, a nonlinear finite element approach would normally be the method of choice for developing simulations. However, the computing cost for these procedures is extremely high, and they may fail to converge at large strains (for example, 30%), so they are generally neither efficient nor stable for real-time surgery simulation. With the interpolation approach, we are able to overcome the stability problem at large strains and achieve significant improvements in speed. Consequently, we are able to provide high fidelity simulations for surgery planning and training, as well as a real time tool for correcting morphological changes during interventional procedures such as brachytherapy.

Even though the material nonlinearity addressed in this paper is typical to soft tissue in general, many applications often involve large deformation but small strains as in the case of rotation, so the geometric nonlinearity in the matrix B of equation (4) has to be taken into consideration. Otherwise, significant accuracy may be lost. In addition, while we have demonstrated the speed and accuracy of the polynomial approximation for a few contacted nodes, we have yet to demonstrate that this approach is appropriate for a large contact area, involving most of the nodes on the surface. In future work, we will address these problems as well as other multi-dimensional interpolation issues.

References

- ABAQUS. Analysis Users Manual, version 6.4. ABAQUS, Inc.
- D. Baraff and A. Witkin, "Large steps in cloth simulation", in *Computer Graphics (SIGGRAPH 98 Conference Proceedings)* M. Cohen, Ed., Florida: Orlando, 1998, pp. 43–54.
- J. Berkley, G. Turkiyyah, D. Berg, M. Ganter and S. Weghorst, "Real-time finite element modeling for surgery simulation: an application to virtual suturing", *IEEE Trans. Vis. Comput. Graph.*, 10, pp. 314–325, 2004.
- S.D. Bona, L. Lutzemberger and O. Salvetti, "A simulation model for analyzing brain structure deformation", *Phys. Med. Biol.*, 48, pp. 4001–4022, 2003.
- M. Bro-Nielsen and S. Cotin, "Real-time volumetric deformable models for surgery simulation using finite element and condensation", *Computer Graphics Forum, Eurographics 96*, 1996, pp. 57–66

- M. Chabanas, Y. Payan, C. Marecaux, P. Swider and F. Boutault, *Comparison of Linear and Non-linear Soft Tissue Models with Post-operative CT Scan in Maxillofacial Surgery*, S. Cotin and D.N. Metaxas, Eds., LNCS3078, Springer, 2004, pp. 19–27.
- S. Cotin, H. Delingette and N. Ayache, “Real-time elastic deformations of soft tissues for surgery simulation”, *IEEE Trans. Vis. Comput. Graph.*, 5, pp. 62–73, 1999.
- S. Cotin, H. Delingette and N. Ayache, “A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation”, *Vis. Comput.*, 16, pp. 437–452, 2000.
- H. Delingette, “Toward realistic soft-tissue modeling in medical simulation”, *Proc. IEEE*, 86, pp. 512–523, 1998.
- J.G. Easley, *Mechanics of Elastic Structures*, Prentice-Hall, 1989.
- Y.B. Fu and R.W. Ogden, *Nonlinear Elasticity: Theory and Applications*, Cambridge University Press, 2001.
- Y.C. Fung, *Mechanical Properties of Living Tissues*, New York: Springer-Verlag, 1981.
- S.F.F. Gibson, “3d chainmail: a fast algorithm for deforming volumetric objects”, *1997 Symposium on Interactive 3D Graphics*, Providence, Rhode Island, 1997, pp. 149–154.
- L. Glass, P.J. Hunter and A. McCulloch, *Theory of Heart Biomechanics, Biophysics, and Nonlinear Dynamics of Cardiac Function*, New York: Springer-Verlag, 1991.
- W.W. Hager, “Updating the Inverse of a matrix”, *SIAM Rev.*, 31, pp. 221–239, 1989.
- G.A. Holzapfel and R.W. Ogden, *Biomechanics of soft Tissue in Cardiovascular Systems*, New York: Springer-Verlag, 2003.
- D.L. James and D.K. Pai, *Accurate Real Time Deformable Objects. Computer Graphics (SIGGRAPH 99 Conference Proceedings)*, Los Angeles, California, 1999, pp. 65–72.
- Y. Lee, D. Terzopoulos and K. Walters, “Realistic modeling for facial animation”, in *Computer Graphics (SIGGRAPH 95 Conference Proceedings)*, S.G. Mair and R. Cook, Eds., Orlando, Florida, 1995, pp. 55–62.
- Y. Liu, A.E. Kerdok and R.D. Howe, *A Nonlinear Finite Element Model of Soft Tissue Indentation*, Cambridge, MA: Springer-Verlag, 2004, pp. 67–76.
- W.E. Lorensen and H.E. Cline, “Marching cubes: a high resolution 3D surface construction algorithm”, *ACM Comput. Graph.*, pp. 163–169, 1987.
- D.F. Meaney, “Relationship between structural modeling and hyper-elastic material behavior: application to CNS white matter”, *Biomech. Model Mechanobiol.*, April, 1(4), Springer-Verlag, pp. 279–293, 2003.
- H.W. Nienhuys and A.F. van der Stappen, “Combining finite element deformation with cutting for surgery simulation”, in *Proceedings of Eurographics 2000*, A. Sousa and J.C. de Torres, Eds., 2000, pp. 143–152.
- H.W. Nienhuys, *Cutting in Deformable Objects*, PhD Thesis, University Utrecht, 2003.
- R.W. Ogden, *Non-linear Elastic Deformations*, New York: Ellis Horwood, 1984.
- P. Pathmanathan, D. Gavaghan, J. Whiteley, M. Brady, M. Nash, P. Nielsen and V. Rajagopal, Predicting tumour location by simulating large deformations of the breast using a 3D finite element model and nonlinear elasticity, in *Proceeding of MICCAI2004*, B. Christian, D.R. Haynor and P. Hellier, Eds., LNCS3217, Springer-Verlag, 2004, pp. 217–224.
- S.H. Peng and W.V. Chang, “A compressible approach in finite element analysis of rubber-elastic materials”, *Comp. Struct.*, 62(3), pp. 573–593, 1997.
- G. Picinbono, H. Delingette and A. Ayache, “Non-linear and anisotropic elastic soft tissue models for medical simulation”, *IEEE International Conference Robotics and Automation*, 2001, pp. 1370–1375, Seoul, Korea.
- W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, New York: Cambridge University Press, 1988.
- A. Samani, J. Bishop, E. Ramsay and D.B. Plewes, “A 3-D contact problem finite element model for breast shape deformation derived from MRI data”, Presented at the 23rd Annual Meeting of the American Society of Biomechanics, University of Pittsburgh, 1999.
- D. Terzopoulos and K. Fleischer, “Modeling inelastic deformation: Viscoelasticity, plasticity, fracture”, in *Computer Graphics (SIGGRAPH 88 Conference Proceedings)*, J. Dill, Ed., Atlanta, Georgia, 1988, pp. 269–278.
- D. Terzopoulos and K. Waters, “Physically-based facial modeling, analysis and animation”, *J.Visua. Compu. Anima*, 1, pp. 73–80, 1990.
- T.P. Usyk and A. McCulloch, “Computational methods for soft tissue Biomechanics”, in G.A. Holzapfel and R.W. Ogden, Eds., *Biomechanics of Soft Tissue in Cardiovascular Systems*, New York: Springer-Verlag, 2003, pp. 273–341.
- X. Wu, T. Goktekin and F. Tendrick, “Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes”, *Proc. Eurographics 2001*, 2001, pp. 349–358.
- H. Zhong, M.P. Wachowiak and T.M. Peters, “Adaptive finite element technique for cutting in surgical simulation”, *Proc. SPIE Med. Imaging*, 5744, 2005.
- Y. Zhuang and J. Canny, “Haptic interaction with global deformations”, *Proc. ICRA 2000 (IEEE International Conference on Robotics and Automation)*, San Francisco, USA, 2000, pp. 2428–2433.
- O.C. Zienkiewicz and R.L. Taylor, *The Finite Element Method, Volume 1: The Basis Barcelona*, Spain: Butterworth Heinemann, 2000.
- O.C. Zienkiewicz and R.L. Taylor, *The Finite Element Method, Volume 2: Solid Mechanics Barcelona*, Spain: Butterworth Heinemann, 2000.