

# Robust Quasistatic Finite Elements and Flesh Simulation

Joseph Teran<sup>†</sup>  
Stanford University

Eftychios Sifakis<sup>‡</sup>  
Stanford University  
Intel Corporation

Geoffrey Irving<sup>‡</sup>  
Stanford University  
Pixar Animation Studios

Ronald Fedkiw<sup>‡</sup>  
Stanford University  
Industrial Light+Magic

---

## Abstract

*Quasistatic and implicit time integration schemes are typically employed to alleviate the stringent time step restrictions imposed by their explicit counterparts. However, both quasistatic and implicit methods are subject to hidden time step restrictions associated with both the prevention of element inversion and the effects of discontinuous contact forces. Furthermore, although fast iterative solvers typically require a symmetric positive definite global stiffness matrix, a number of factors can lead to indefiniteness such as large jumps in boundary conditions, heavy compression, etc. We present a novel quasistatic algorithm that alleviates geometric and material indefiniteness allowing one to use fast conjugate gradient solvers during Newton-Raphson iteration. Additionally, we robustly compute smooth elastic forces in the presence of highly deformed, inverted elements alleviating artificial time step restrictions typically required to prevent such states. Finally, we propose a novel strategy for treating both collision and self-collision in this context.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling – Physically based modeling; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism – Animation

---

## 1. Introduction

Fast and robust simulations of elastic solids are becoming increasingly important in computer graphics applications due largely to the prominence of virtual characters. Feature films such as Van Helsing, Spiderman, The Lord of the Rings and countless others benefit from the use of humanoid characters in scenes that would be difficult and expensive if not impossible to create with live actors, see e.g. [KMGB04, ST04]. Typical models are composed of an underlying skeleton whose motion is prescribed kinematically (from motion capture or traditional animation) and a mechanism for transmitting the skeletal motion to skin deformation. Physics based simulations of musculature and fleshy tissues are becoming increasingly popular for producing these deformations, especially when virtual characters undergo contact and collision with the surrounding environment. Moreover, faithfully depicting the artist's conception of the character requires reasonably high resolution tetrahedral meshes placing additional demands for efficiency on the simulation algorithm.

Since explicit time integration schemes can often have stringent time step restrictions, various authors have investigated

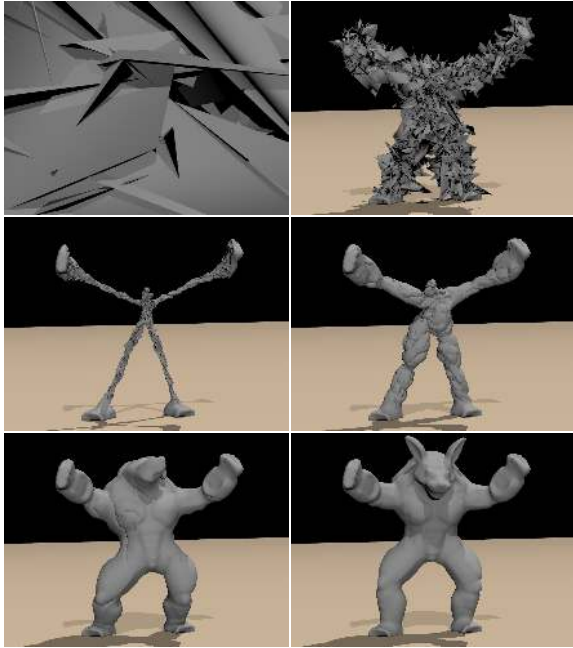
the use of semi-implicit (e.g. [BMF03]), fully implicit (e.g. [TF88b, BW98]) and quasistatic (e.g. [HFS\*01, MMDJ01]) time integration schemes. Quasistatic schemes ignore inertial effects and thus are not suitable for simulating less constrained phenomena such as ballistic motion. However, in applications where inertial effects are relatively small compared to the deformation caused by contact, collision, and time varying boundary conditions, quasistatic solvers can often provide a speedup of one to two orders of magnitude over explicit schemes. For example, quasistatic simulations are well suited for flesh deformation where the flesh is rigidly attached to bones and heavily influenced by contact, collision and self-collision.

Although implicit and quasistatic schemes remove the time step restriction associated with wave propagation, the Newton-Raphson method used to solve the resulting nonlinear equations may produce inverted elements during iteration when large time steps are used, bringing the algorithm to a halt. For example, large displacement boundary conditions tend to invert elements unless steps are taken to distribute the effects to surrounding elements, and the typical approach is to impose an artificial time step restriction even in the quasistatic case. This has been discussed in both the computer graphics (e.g. [HFS\*01]) and the computational physics (e.g. [GW03]) literature. Even in the case where the

---

<sup>†</sup> email: jteran@stanford.edu

<sup>‡</sup> email: {sifakis|irving|fedkiw}@cs.stanford.edu



**Figure 1:** The particles of a tetrahedron mesh are randomly scattered across a tenfold magnification of its bounding box and the object is subsequently evolved to steady state using our robust quasistatic solver. From top to bottom and left to right the Newton iteration counts are 0, 1, 2, 10, 40, and 80. The hands and feet are specified as boundary conditions.

final mesh will be inversion free, artificially small time steps are required to ensure that every intermediate state considered during Newton-Raphson iteration is also inversion free restricting the speed at which one can converge to the desired solution. Recently, researchers have aimed at handling inversion using altitude springs [MBTF03], volume preservation terms [THMG04], rotated linear models [MG04], etc. However, these methods change or limit the underlying partial differential equation, whereas [ITF04] allows for general nonlinear constitutive models with forces that are smooth enough to be used in conjunction with iterative methods. Thus, we adopt the approach of [ITF04] and extend it to the quasistatic regime removing the artificial time step restriction required by other schemes making our solution method extremely efficient.

In each Newton-Raphson iteration, the nonlinear system of equations is reduced to a linear system that must be solved to advance to the next iteration. This linear system is guaranteed to be symmetric and positive definite in the vicinity of equilibrium states, enabling the use of fast conjugate gradient solvers. Unfortunately, the use of large time steps produces substantial divergence from a steady state, leading to a symmetric linear system that is often indefinite. State of the art finite element packages such as NIKE3D still use direct solvers such as that proposed in [TWS80], even though such methods are much slower and require considerably more memory than iterative methods. [GW03] first

try a fast iterative solver switching to a slower direct method when it fails. [HFS\*01] discussed these issues in the context of quasistatic simulation pointing out the erratic behavior of conjugate gradient methods and a preference against direct methods. By adding an artificial “viscosity” to their simulations, they were able to obtain reasonable results with a GMRES iterative scheme. In the context of implicit time integration, [CK02] pointed out that extra damping forces such as those applied in [BW98, VT00] can help to overcome indefiniteness, but not guarantee it. Furthermore, they point out that this damping degrades the realism of the simulation. Instead, they take a closer look at the problem in the case of springs identifying compression as a source of indefiniteness and proposing a technique to guarantee definiteness in the special case of cloth simulation with springs. A key contribution of our paper is a new and general method for guaranteeing positive definiteness, thus allowing for the use of fast conjugate gradient solvers under all circumstances (including inversion) for arbitrary constitutive models in the finite element framework. Our method modifies the search path followed towards equilibrium without altering the set of equilibrium solutions or the governing equations.

## 2. Previous Work

[TPBF87, TF88b, TF88a] pioneered deformable models in computer graphics including early work on plasticity and fracture. Finite element simulations have been used to model a hand grasping a ball [GMTT89], for virtual surgery [PDA01], fracture [OH99, MMDJ01, OBH02, MBF04], etc. Other work includes the adaptive frameworks of [DDCB01, GKS02, CGC\*02b], the rotation based approaches in [MDM\*02, MG04, CK05] (see also [TW88]), the bending models in [BMF03, GHDS03], the precomputed data driven models of [JF03], and the point based methods in [MKN\*04].

The construction of muscles and/or flesh deformation is important for computer graphics characters, and anatomy based modeling techniques of varying resolutions have been applied. [WV97, SPCM97] used anatomically based models of muscles, tendons and fatty tissue to deform an outer skin layer. [NTHF02] fit deformable B-spline solids to anatomic data in order to create volumetric, anisotropic representations of muscles and their internal structures. [AHS03] used a variety of techniques to model a human hand. More biomechanically accurate techniques for muscle simulation were proposed in [CZ92, HFS\*01, TBNF03], and a number of researchers are working to simulate data from the NIH visible human dataset, e.g. [ZCK98, HFS\*01, DCKY02, TBNF03].

Instead of creating an explicit model for muscle and fatty tissue, one can place an articulated skeleton inside the character skin and formulate correspondences between each vertex on the skin mesh and the various joints in the skeleton. This is typically called enveloping or skinning and can suffer from a number of artifacts especially near joints such as

elbows and shoulders. A number of techniques have been proposed to overcome these difficulties, see for example [LCF00, SRC01, WP02, MG03]. [ACP02] used these techniques in conjunction with range scan data, and [KM04] used them to model a human hand. [KJP02] proposed a similar method that used principal component analysis and a library of deformations precomputed with nonlinear static finite element analysis. Although these techniques are fast and do not require one to build an underlying muscle model for each character, they can lead to lower quality results than full finite element simulations. A physically based approach was taken in [JP02] to add ballistic motion to character skins in otherwise kinematically constructed motions. [CGC\*02a] approaches this problem by embedding the character in a coarse finite element mesh which deforms rigidly with the bones, but obeys a linear finite element model locally to each bone.

### 3. Quasistatic Formulation

Using Newton's second law of motion we can describe the evolution of a deformable body using the equations  $\ddot{\mathbf{x}}_t = \ddot{\mathbf{v}}$  and  $\ddot{\mathbf{v}}_t = M^{-1}\ddot{\mathbf{f}}(t, \mathbf{x}, \mathbf{v})$  where  $\mathbf{x}$ ,  $\mathbf{v}$  and  $\mathbf{f}$  denote the positions, velocities and aggregate forces of *all* the nodes of the tetrahedral mesh. (We use  $\mathbf{x}$  as the vector valued position of a single node.)  $M$  is the mass matrix, which is diagonal in our lumped mass formulation. The nodal forces can be decomposed into internal and external forces,  $\vec{\mathbf{f}} = \vec{\mathbf{f}}_{int} + \vec{\mathbf{f}}_{ext}$ , the latter being supplied as time varying input to the simulation.

We apply a quasistatic assumption that both the accelerations and velocities are zero to obtain  $\vec{\mathbf{f}}(t, \mathbf{x}, \vec{\mathbf{0}}) = \vec{\mathbf{0}}$  which states that the externally supplied time varying input must be balanced by the internal resistance of the material. In particular, we use a nonlinear finite element method to solve for the internal forces, and thus we must invert a nonlinear equation to find the time varying positions  $\mathbf{x}(t)$  at any time  $t$ . This is accomplished with a Newton-Raphson iterative solver, and each step towards the steady state solution begins with the linearization of the nodal forces about the current solution estimate  $\vec{\mathbf{x}}_k$ , i.e.  $\vec{\mathbf{f}}(\vec{\mathbf{x}}_k + \Delta\vec{\mathbf{x}}_k) \approx \vec{\mathbf{f}}(\vec{\mathbf{x}}_k) + \left(\frac{\partial\vec{\mathbf{f}}}{\partial\vec{\mathbf{x}}}\right)_{\vec{\mathbf{x}}_k} \Delta\vec{\mathbf{x}}_k$  where  $\Delta\vec{\mathbf{x}}_k = \vec{\mathbf{x}}_{k+1} - \vec{\mathbf{x}}_k$ . Since we desire force equilibrium with  $\vec{\mathbf{f}}(\vec{\mathbf{x}}_{k+1}) = \vec{\mathbf{f}}(\vec{\mathbf{x}}_k + \Delta\vec{\mathbf{x}}_k) = \vec{\mathbf{0}}$ , we solve the linear system

$$-\left.\frac{\partial\vec{\mathbf{f}}}{\partial\vec{\mathbf{x}}}\right|_{\vec{\mathbf{x}}_k} \Delta\vec{\mathbf{x}}_k = \vec{\mathbf{f}}(\vec{\mathbf{x}}_k) \quad (1)$$

to find the next iterate  $\vec{\mathbf{x}}_{k+1}$ .

Although the quasistatic assumption does not apply to free falling, unconstrained, lightly damped objects whose richness of deformation is largely enhanced by the effects of inertia, it is a viable modeling strategy for a range of applications in which boundary conditions and external forces predominantly determine the material state (e.g. skeletal muscles under a variety of conditions).

### 4. Strain Energy

For a hyperelastic material, the nodal forces can be defined via the energy as  $\vec{\mathbf{f}} = -\partial\Psi/\partial\vec{\mathbf{x}}$ , and thus we can rewrite equation (1) as

$$\left.\frac{\partial^2\Psi}{\partial\vec{\mathbf{x}}^2}\right|_{\vec{\mathbf{x}}_k} \Delta\vec{\mathbf{x}}_k = -\left.\frac{\partial\Psi}{\partial\vec{\mathbf{x}}}\right|_{\vec{\mathbf{x}}_k}. \quad (2)$$

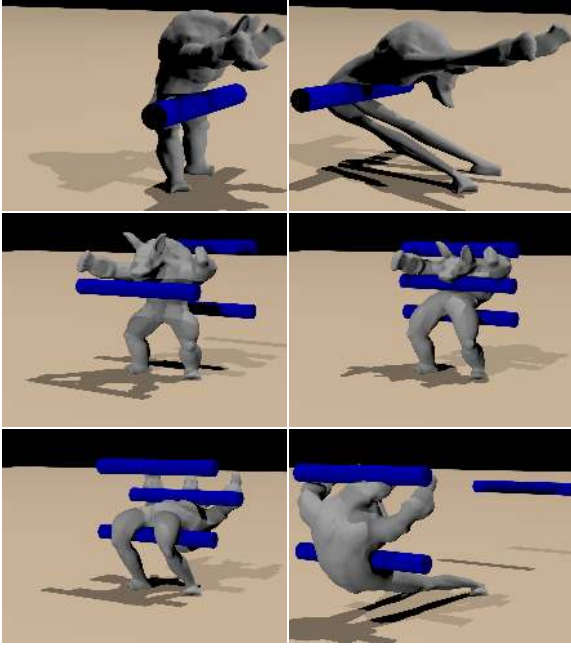
That is, the global stiffness matrix  $-\partial\vec{\mathbf{f}}/\partial\vec{\mathbf{x}}$  is *always* symmetric, as a result of the hyperelastic energy having continuous second derivatives with respect to the spatial configuration. Furthermore, a steady state corresponds to a local minimum of the hyperelastic energy indicating that the energy Hessian,  $\partial^2\Psi/\partial\vec{\mathbf{x}}^2$ , (or equivalently the global stiffness matrix) is positive definite in the vicinity of an *isolated* steady state. Moreover, systems that possess steady states along a continuous manifold in configuration space, such as underconstrained bodies with rigid degrees of freedom (e.g. a single spring with only one fixed endpoint that is otherwise free to rotate), still exhibit semi-definite stiffness matrices at their steady state. Thus, such systems can be reduced to the fully constrained case by factoring out the manifold of the configuration space that does not affect the hyperelastic energy.

Symmetry of the coefficient matrix in the linear system (2) allows for the use of symmetric solvers, and direct methods are commonly used. However, the fact that the stiffness matrix is positive definite close to the steady state suggests that symmetric positive definite solvers such as the conjugate gradient method *might* be applicable. This would alleviate the drawbacks of direct methods including the need to explicitly form the stiffness matrix, the memory demands incurred by matrix fill during the direct solve, and the excessive computational expense of direct solvers as opposed to iterative methods.

Our method modifies the coefficient matrix in equation (2) into a *positive definite* symmetric matrix and proceeds to compute the next iterate  $\Delta\vec{\mathbf{x}}_k$  using this modified system. We emphasize that this modification only alters individual steps towards a minimum of the strain energy and not those minima themselves. These modifications are localized to regions of the simulation mesh that contribute to this indefiniteness. This practice of modifying the Hessian of the optimization functional is common in the optimization literature (see e.g. [GMW81]) and is usually referred to as a modified Newton method.

### 5. Finite Element Forces

We follow the notation of [TBNF03], and their geometric interpretation of the finite element method. Consider a time dependent map  $\phi$  from the undeformed material coordinates  $\mathbf{X}$  to world coordinates  $\mathbf{x}$ . The stress at a point  $\mathbf{X}$  in the material depends on the deformation gradient  $\mathbf{F}(\mathbf{X}) = \partial\mathbf{x}/\partial\mathbf{X}$  of this mapping. We use constant strain tetrahedral elements where  $\mathbf{F}$  is a constant  $3 \times 3$  matrix in each tetrahedron. We



**Figure 2:** Illustration of large deformation in conjunction with collision. The hands and feet are set as boundary conditions for the first row, but only the feet are fixed for the middle and bottom rows.

define edge vectors for each tetrahedron in both material coordinates,  $\mathbf{d}_{m_1} = \mathbf{X}_1 - \mathbf{X}_0$ ,  $\mathbf{d}_{m_2} = \mathbf{X}_2 - \mathbf{X}_0$ ,  $\mathbf{d}_{m_3} = \mathbf{X}_3 - \mathbf{X}_0$ , and world coordinates,  $\mathbf{d}_{s_1} = \mathbf{x}_1 - \mathbf{x}_0$ ,  $\mathbf{d}_{s_2} = \mathbf{x}_2 - \mathbf{x}_0$ ,  $\mathbf{d}_{s_3} = \mathbf{x}_3 - \mathbf{x}_0$ , and construct  $3 \times 3$  matrices  $\mathbf{D}_m$  and  $\mathbf{D}_s$  using the edge vectors as columns. Then  $\mathbf{F} = \mathbf{D}_s \mathbf{D}_m^{-1}$ , and  $\mathbf{D}_m^{-1}$  is constant and can be precomputed and stored for efficiency.

For hyperelastic materials, stress is defined as the derivative of a strain energy typically constructed from various strain invariants, and we use the first Piola-Kirchhoff stress which is the gradient of the strain energy with respect to the deformation gradient,  $\mathbf{P} = \partial\Psi/\partial\mathbf{F}$ .  $\mathbf{P}$  maps area weighted normals in material space to forces in world space. The force on a node  $i$  due to a single tetrahedron incident to it is  $\mathbf{g}_i = -\mathbf{P}(A_1\mathbf{N}_1 + A_2\mathbf{N}_2 + A_3\mathbf{N}_3)/3$ , where the  $A_j\mathbf{N}_j$  are the area weighted normals of the faces of the tetrahedron incident to node  $i$ . Since these do not change during the simulation, we can precompute a vector  $\mathbf{b}_i$  such that  $\mathbf{g}_i = \mathbf{P}\mathbf{b}_i$ . For efficiency, we compute  $\mathbf{g}_0 = -(\mathbf{g}_1 + \mathbf{g}_2 + \mathbf{g}_3)$  and compactly express the other three  $\mathbf{g}_i$  as  $\mathbf{G} = \mathbf{P}\mathbf{B}_m$  where  $\mathbf{G} = (\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3)$  and  $\mathbf{B}_m = (\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3) = -V\mathbf{D}_m^{-T}$  with  $V$  the volume of the tetrahedron in material space.

As noted in [ITF04], the first Piola-Kirchhoff stress is invariant under rotations of either material or world space for isotropic materials. Furthermore, the deformation gradient can be transformed into a diagonal matrix,  $\hat{\mathbf{F}}$ , with an application of a material and a world space rotation,  $\mathbf{F} = \mathbf{U}\hat{\mathbf{F}}\mathbf{V}^T$ . This decomposition is obtained from the standard singular value decomposition of  $\mathbf{F}$  along with the subsequent removal

of any reflections in the orthogonal  $\mathbf{U}$  and  $\mathbf{V}$ . This requires the negation of the smallest singular value of  $\hat{\mathbf{F}}$  for inverted tetrahedra. Combining the rotational invariance of the first Piola-Kirchhoff stress with the diagonalization of the deformation gradient yields

$$\mathbf{P}(\mathbf{F}) = \mathbf{U}\mathbf{P}(\mathbf{U}^T\mathbf{F}\mathbf{V})\mathbf{V}^T = \mathbf{U}\mathbf{P}(\hat{\mathbf{F}})\mathbf{V}^T \quad (3)$$

where  $\mathbf{P}(\hat{\mathbf{F}})$  is also diagonal for isotropic materials. This factorization is particularly convenient, because it allows for a simple extension of the constitutive model to inverted elements in a smooth manner. That is, one only needs to modify the diagonal  $\mathbf{P}(\hat{\mathbf{F}})$  to be valid for a single negative entry in the diagonal  $\hat{\mathbf{F}}$ . For more details, see [ITF04].

## 6. Element Stiffness Matrix

The global stiffness matrix in equation (1) is constructed from the additive contributions of the element stiffness matrices,  $-\partial\mathbf{f}/\partial\mathbf{x}$ , which are based on contributions from individual tetrahedra. As a result of this additive decomposition, definiteness of the element stiffness matrices is a sufficient condition for definiteness of the global stiffness matrix. Motivated by this fact, we manipulate the element stiffness matrix to ensure global definiteness. In section 8 we show that this elemental manipulation amounts to the solution of a single  $3 \times 3$  symmetric eigenproblem and a few simple algebraic operations. In contrast, dealing with the global stiffness matrix directly can be prohibitively expensive, especially if eigenanalysis or Cholesky factorization of that matrix is required, as in most standard approaches to treating locally indefinite optimization problems [GMW81].

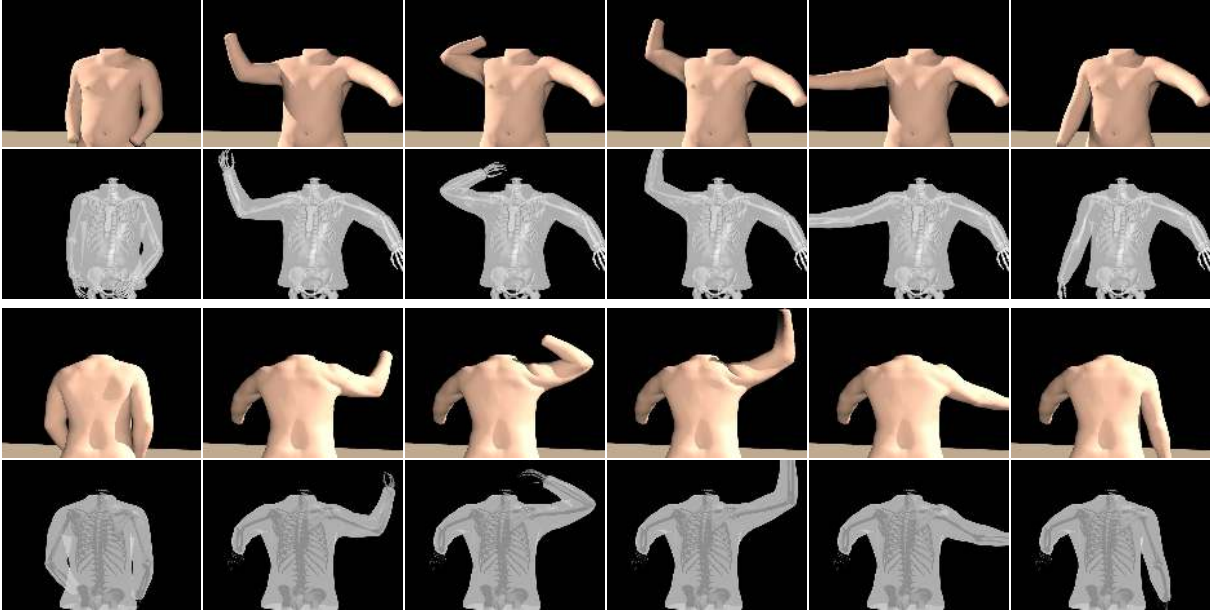
In order to establish the positive definiteness of the element stiffness matrix, we must ensure that  $\delta\mathbf{x}^T(-\partial\mathbf{f}/\partial\mathbf{x})\delta\mathbf{x} = -\delta\mathbf{x}^T\delta\mathbf{f} > 0$  for any increment  $\delta\mathbf{x}$ . Using the formulas from the last section and some tensor manipulations yields

$$\begin{aligned} \delta\mathbf{x}^T\delta\mathbf{f} &= \sum_{i=1}^3 \delta\mathbf{x}_i^T \delta\mathbf{g}_i - \delta\mathbf{x}_0^T \sum_{i=1}^3 \delta\mathbf{g}_i = \sum_{i=1}^3 (\delta\mathbf{x}_i - \delta\mathbf{x}_0)^T \delta\mathbf{g}_i \\ &= \delta\mathbf{D}_s : \delta\mathbf{G} = \text{tr}[\delta\mathbf{D}_s^T \delta\mathbf{G}] = -V\text{tr}[\delta\mathbf{D}_s^T \delta\mathbf{P}\mathbf{D}_m^{-T}] \\ &= -V\text{tr}[\mathbf{D}_m^{-T} \delta\mathbf{D}_s^T \delta\mathbf{P}] = -V\text{tr}[\delta\mathbf{F}^T \delta\mathbf{P}] = -V(\delta\mathbf{F} : \delta\mathbf{P}). \end{aligned}$$

Since the material element volume  $V$  is always a positive constant, the positive definiteness condition reduces to  $\delta\mathbf{F} : \delta\mathbf{P} > 0$  or  $\delta\mathbf{F} : (\partial\mathbf{P}/\partial\mathbf{F}) : \delta\mathbf{F} > 0$ . Therefore, the positive definiteness of the element stiffness matrix is equivalent to the positive definiteness of the fourth order tensor  $\partial\mathbf{P}/\partial\mathbf{F}$ . This result is in direct analogy with the energy based formulation of the Newton-Raphson iteration system (2), since by definition  $\mathbf{P} = \partial\Psi/\partial\mathbf{F}$  and thus  $\partial\mathbf{P}/\partial\mathbf{F} = \partial^2\Psi/\partial\mathbf{F}^2$ .

## 7. Diagonalization

Testing and enforcing positive definiteness of the fourth order tensor  $\partial\mathbf{P}/\partial\mathbf{F}$  directly can be rather unwieldy. Instead,



**Figure 3:** Simulation of quasistatic flesh deformation driven by a kinematic skeleton.

we start as in [ITF04] by rotating both stresses and deformations into diagonal space (transforming our configuration using the rotation matrices that diagonalize the current  $\mathbf{F}$  and  $\mathbf{P}$ ). In order to do this, first note that  $\delta\mathbf{P} = (\partial\mathbf{P}(\mathbf{F})/\partial\mathbf{F})|_{\mathbf{F}} : \delta\mathbf{F}$  where we explicitly stress the dependency of  $\mathbf{P}$  on  $\mathbf{F}$  with  $\mathbf{P}(\mathbf{F})$ . We can manipulate this equality into

$$\begin{aligned} \delta\mathbf{P} &= \frac{\partial\mathbf{U}\mathbf{P}(\mathbf{U}^T\mathbf{F}\mathbf{V})\mathbf{V}^T}{\partial(\mathbf{U}^T\mathbf{F}\mathbf{V})} \Big|_{\mathbf{F}} : \delta(\mathbf{U}^T\mathbf{F}\mathbf{V}) \\ &= \mathbf{U} \left\{ \frac{\partial\mathbf{P}(\mathbf{F})}{\partial\mathbf{F}} \Big|_{\mathbf{U}^T\mathbf{F}\mathbf{V}} : \mathbf{U}^T \delta\mathbf{F}\mathbf{V} \right\} \mathbf{V}^T \\ &= \mathbf{U} \left\{ \frac{\partial\mathbf{P}}{\partial\mathbf{F}} \Big|_{\hat{\mathbf{F}}} : \mathbf{U}^T \delta\mathbf{F}\mathbf{V} \right\} \mathbf{V}^T \end{aligned} \quad (4)$$

where the first equality comes from equation (3) and replacing  $\delta\mathbf{F}$  with a rotated version, the second comes from a change of variables and the fact that  $\mathbf{U}$  and  $\mathbf{V}$  are chosen independent of  $\mathbf{F}$ , and the third comes from choosing  $\mathbf{U}$  and  $\mathbf{V}$  to be the rotation matrices that diagonalize the initial value of  $\mathbf{F}$ , i.e. where we evaluate  $\partial\mathbf{P}/\partial\mathbf{F}$  to linearize for iteration. Also in the last equality, we drop the explicit dependence of  $\mathbf{P}$  on  $\mathbf{F}$ .

Equation (4) provides all the information we need for solving the Newton-Raphson iteration system using a conjugate gradient solver, since the nodal force differentials can readily be computed from the stress differentials as  $\delta\mathbf{G} = \delta\mathbf{P}\mathbf{B}_m$ . Furthermore we have

$$\begin{aligned} \delta\mathbf{P} : \delta\mathbf{F} &= \mathbf{U} \left\{ \frac{\partial\mathbf{P}}{\partial\mathbf{F}} \Big|_{\hat{\mathbf{F}}} : \mathbf{U}^T \delta\mathbf{F}\mathbf{V} \right\} \mathbf{V}^T : \delta\mathbf{F} \\ &= \mathbf{U}^T \delta\mathbf{F}\mathbf{V} : \frac{\partial\mathbf{P}}{\partial\mathbf{F}} \Big|_{\hat{\mathbf{F}}} : \mathbf{U}^T \delta\mathbf{F}\mathbf{V} \end{aligned}$$

illustrating that the condition for definiteness,  $\delta\mathbf{P} : \delta\mathbf{F} > 0$ ,

derived in section 6 is equivalent to positive definiteness of  $(\partial\mathbf{P}/\partial\mathbf{F})|_{\hat{\mathbf{F}}}$ . We might expect that applying the rotations that diagonalize the current deformation  $\mathbf{F}$  to  $\delta\mathbf{P}$  and  $\delta\mathbf{F}$  would induce a simple structure for the tensor  $(\partial\mathbf{P}/\partial\mathbf{F})|_{\hat{\mathbf{F}}}$ . In fact this tensor turns out to have a block diagonal structure in the case of isotropic materials.

## 8. Enforcing Positive Definiteness

In order to reveal the block diagonal structure of  $(\partial\mathbf{P}/\partial\mathbf{F})|_{\hat{\mathbf{F}}}$ , we rewrite the  $3 \times 3 \times 3 \times 3$  fourth order tensor as a  $9 \times 9$  matrix. To do this, we consider the rearrangement of a  $3 \times 3$  matrix  $\mathbf{S}$  into the  $9 \times 1$  vector  $(s_{11}, s_{22}, s_{33}, s_{12}, s_{21}, s_{13}, s_{31}, s_{23}, s_{32})$ . We can then represent  $(\partial\mathbf{P}/\partial\mathbf{F})|_{\hat{\mathbf{F}}}$  as the  $9 \times 9$  matrix that maps the vector equivalent of  $\delta\mathbf{F}$  to the vector equivalent of  $\delta\mathbf{P}$ . For isotropic materials this matrix is block diagonal with diagonal components  $\mathbf{A}$ ,  $\mathbf{B}_{12}$ ,  $\mathbf{B}_{13}$  and  $\mathbf{B}_{23}$  where

$$\mathbf{A} = \begin{bmatrix} \alpha_{11} + \beta_{11} + \gamma_{11} & \gamma_{12} & \gamma_{13} \\ \gamma_{12} & \alpha_{22} + \beta_{22} + \gamma_{22} & \gamma_{23} \\ \gamma_{13} & \gamma_{23} & \alpha_{33} + \beta_{33} + \gamma_{33} \end{bmatrix}, \quad \mathbf{B}_{ij} = \begin{bmatrix} \alpha_{ij} & \beta_{ij} \\ \beta_{ij} & \alpha_{ij} \end{bmatrix}$$

Here,

$$\alpha_{ij} = 2\Psi_I + 4(\sigma_i^2 + \sigma_j^2)\Psi_{II}$$

$$\beta_{ij} = 4\sigma_i\sigma_j\Psi_{II} - \frac{2III\Psi_{III}}{\sigma_i\sigma_j}$$

$$\gamma_{ij} = \left( 2\sigma_i \ 4\sigma_i^3 \ \frac{2III}{\sigma_i} \right) \frac{\partial^2\Psi}{\partial(I, II, III)^2} \begin{pmatrix} 2\sigma_j \\ 4\sigma_j^3 \\ \frac{2III}{\sigma_j} \end{pmatrix} + \frac{4III\Psi_{III}}{\sigma_i\sigma_j}$$

where  $\Psi = \Psi(I, II, III)$  is the strain energy written in terms of the invariants  $I = \text{tr } \mathbf{C}$ ,  $II = \mathbf{C} : \mathbf{C}$  and  $III = \det \mathbf{C}$  with  $\mathbf{C} = \mathbf{F}^T \mathbf{F}$  and subscripts representing partial derivatives. Also,  $\sigma_1$ ,  $\sigma_2$  and  $\sigma_3$  are the diagonal components that constitute  $\hat{\mathbf{F}}$ .

Positive definiteness of  $(\partial \mathbf{P} / \partial \mathbf{F})|_{\hat{\mathbf{F}}}$  is equivalent to positive definiteness of each of the blocks  $\mathbf{A}$ ,  $\mathbf{B}_{12}$ ,  $\mathbf{B}_{13}$  and  $\mathbf{B}_{23}$ . For  $\mathbf{A}$  a simple  $3 \times 3$  diagonalization is required, followed by the clamping of all negative eigenvalues to zero. For the  $2 \times 2$  matrices  $\mathbf{B}_{12}$ ,  $\mathbf{B}_{13}$  and  $\mathbf{B}_{23}$  no eigenanalysis is necessary since the negative eigenvalue, if present, can be clamped to zero analytically.

Our algorithm computes the stress differential  $\delta \mathbf{P}$  as outlined in equation (4). First we compute the rotated deformation differential  $\mathbf{U}^T \delta \mathbf{F} \mathbf{V}$ , and then convert this  $3 \times 3$  second order tensor into a  $9 \times 1$  vector and multiply it by the  $9 \times 9$  matrix for  $(\partial \mathbf{P} / \partial \mathbf{F})|_{\hat{\mathbf{F}}}$  to carry out the contraction. Of course, we use the clamped positive definite version of  $(\partial \mathbf{P} / \partial \mathbf{F})|_{\hat{\mathbf{F}}}$ . The result is then converted from a  $9 \times 1$  vector back to a  $3 \times 3$  second order tensor, before being premultiplied by  $\mathbf{U}$  and postmultiplied by  $\mathbf{V}^T$ .

Since we clamp eigenvalues to zero, the element stiffness matrices are only positive semi-definite, not positive definite, which raises the issue of whether the resulting global stiffness matrix could be semi-definite or ill-conditioned itself. In practice, the additive contributions of neighboring elements and boundary conditions always lead to a positive definite global stiffness matrix, even for configurations as extreme as shown in Figure 1. (Note that one *could* clamp to a small positive value as well.) The effect of boundary conditions on the definiteness of the stiffness matrix is analogous to that observed in the matrix resulting from the discretization of the Poisson equation. When all Neumann boundary conditions are specified, the resulting matrix is positive semi-definite. In this case a special version of Conjugate Gradients is still applicable, since an analytic description of the null space is available and, similarly, the global stiffness matrix of an elastic object has a null space corresponding to global translation and linearized rotation. Specification of one or more Dirichlet boundary conditions makes the Poisson matrix strictly positive definite, with positional constraints having the same effect on the definiteness of the global stiffness matrix for elasticity.

## 9. Inverted Elements

Typically, realistic constitutive models have infinite strain energy as the volume of an element approaches zero, and this discourages element inversion when the equations of motion are integrated with a small enough time step to resolve the stiff material response. Nevertheless, each Newton-Raphson iteration of a quasistatic solver begins with a linearization of the elastic forces after which only a finite amount of energy is required to invert the element. In order to efficiently solve

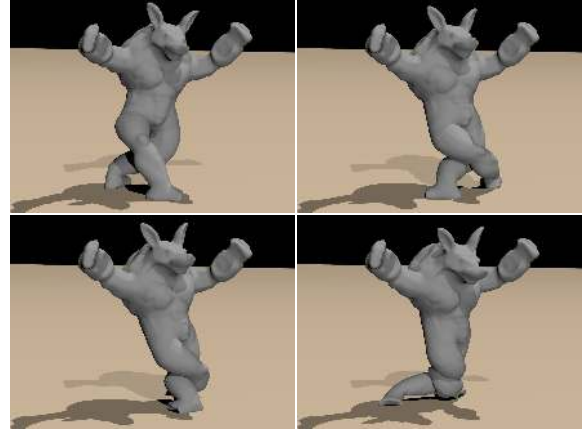


Figure 4: Illustration of self-collision handling.

the equations without artificial limits on the allowable time step, we adopt the approach of [ITF04] smoothly extending the definition of forces past a maximum compression threshold. Constant, linear, or smoother extrapolations can be used for this purpose. In our work constant extrapolation proved to be both simple and sufficient. To implement constant extrapolation we threshold the diagonal values of  $\hat{\mathbf{F}}$  and compute both forces and force differentials using the thresholded deformation gradient. The resulting force differentials are then treated for indefiniteness.

## 10. Collisions

For volumetric collisions one could use the method in [BFA02] applied to the triangulated boundary surface of the tetrahedron mesh as was done in [ITF04]. There is also the self-collision untangling strategy of [BWK03]. But we prefer a penalty based formulation that can more readily be incorporated into the quasistatic formulation. We use a penalty force for collision of our objects with themselves, other deformable tetrahedral bodies and rigid bodies. As a consequence of using penalty forces, the steady state may exhibit slight interpenetration of the colliding surfaces, an effect that is rather subtle and acceptable for our line of applications. The penetration depth can also be adjusted by changing the stiffness of the penalty forces. A penetrating node receives a force in the form of the gradient of a penetration potential defined as  $\Psi_p(\mathbf{x}) = k\phi^2(\mathbf{x})/2$  where  $\phi$  is the signed distance to the surface of the object for  $\mathbf{x}$  interior to the body and zero otherwise. Then the force is  $\mathbf{f}_p = -k\phi(\mathbf{x})\nabla\phi(\mathbf{x})$ , and the force differential is  $\delta \mathbf{f}_p = -k(\nabla\phi(\mathbf{x})\nabla\phi^T(\mathbf{x}) + \phi(\mathbf{x})\partial^2\phi/\partial\mathbf{x}^2|_{\mathbf{x}})\delta\mathbf{x}$ . These forces can corrupt the definiteness of the linearized forces used with Newton-Raphson iteration. The potential for indefiniteness arises from isocontours of the signed distance function with curvatures of differing sign, see e.g. [AS96]. These curvatures are the eigenvalues of  $\partial^2\phi/\partial\mathbf{x}^2$ , and we assure definiteness by projecting this matrix to its positive definite component in the case of rigid body collisions. For deformable

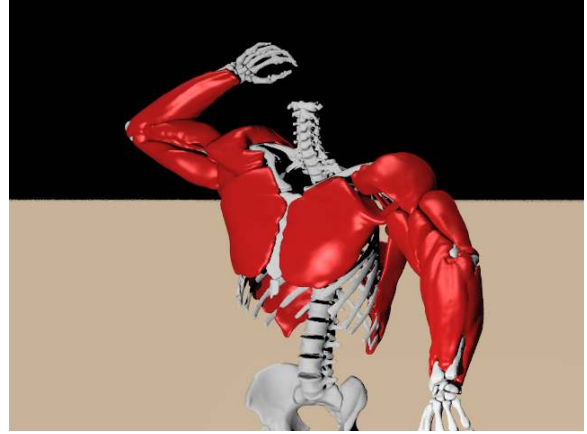
object collisions, we omit the last term altogether. As before, this modification does not change the equilibrium states, only the convergence path towards one of these states.

We take a level set approach (see e.g. [OF02]) to computing penetration depth as did [FL01, MAC04], but instead of updating the level set function as the object deforms we utilize a static level set in material space as in [HFS\*01]. However, many key aspects of our algorithm are significantly different than that proposed in [HFS\*01]. For each rigid and deformable object in the scene, we first precompute a signed distance function on a uniform Cartesian or octree grid as in [GBF03]. This representation is computed in object space for rigid bodies and material space for deformable bodies and is not updated as the simulation progresses. Collecting the depth, normal and curvature information is straightforward for rigid bodies, but we propose a novel approach for deformable tetrahedral bodies.

To compute point collisions against deforming tetrahedral bodies, we maintain a bounding box hierarchy for the tetrahedra in each body. Then for each point, we use this hierarchy to find any tetrahedra that our candidate point may lie inside (inverted tetrahedra are ignored as they represent negative space). For each tetrahedron, we compute the barycentric coordinates of our candidate point to determine if the point is either inside or very close to the tetrahedron in question. We do not require robustness here as this computation is not used to determine whether a point is inside an object, but instead the barycentric coordinates are used to transform the point from world space to material space, i.e. the point is placed in material space keeping its same barycentric coordinates but using the nodal positions of the material space tetrahedron.

Then the material space position of the point is used to query the material space level set to see if the point is inside the object, and if so the local unit normal and level set value are used to estimate the closest point on the surface as  $\mathbf{x}_c = \mathbf{x} - \phi \mathbf{N}$  (where  $\phi$  is negative inside the object). If  $\phi \neq 0$  at  $\mathbf{x}_c$  this equation can be iterated on to find an  $\mathbf{x}_c$  as close to the zero level set as is desired. Before the simulation begins, we also precompute a static bounding box hierarchy for the triangles on the surface of the object, and this is used to find the triangle closest to  $\mathbf{x}_c$  as well as the barycentric coordinates of the point on this triangle closest to  $\mathbf{x}_c$ . Before proceeding, we check to make sure that the local level set value at this point on the triangle,  $\mathbf{x}_t$ , is larger than that at the original point  $\mathbf{x}$ , to ensure that  $\mathbf{x}_t$  is actually farther outside the object than  $\mathbf{x}$ . This keeps us from incorrectly pulling points back towards the object (nonphysical stickiness), because of rasterization errors with the level set function that cause it to have a slightly different approximation to the object surface than as given by surface triangle mesh. Finally, the barycentric coordinates of  $\mathbf{x}_t$  are used to find the corresponding point in world space,  $\mathbf{x}_s$ , on the surface of the deforming object.

In this fashion, we do not use the level set in material space



**Figure 5:** *Quasistatic simulation of the upper torso musculature.*

to push points out of the object, which is important because this is unlikely to give us the proper directions for deformed objects. Instead, we merely use the level set to find a point that is truly on the surface of the object. Then the distance from  $\mathbf{x}_s$  to  $\mathbf{x}$  and the vector pointing between them are used to compute  $\phi(\mathbf{x})$  and  $\nabla\phi(\mathbf{x})$  for the penalty forces and differentials.

## 11. Examples

We demonstrate the applicability of our quasistatic algorithm in a number of complex scenarios. To illustrate the robustness of the extension of the elastic response to degenerate and inverted elements, we solve for elastic equilibrium with an armadillo mesh whose vertices are initially randomly distributed on a cube ten times the size of the armadillo mesh itself and whose hands and feet are constrained. Figure 1 shows a number of iterates in the solution process towards equilibrium. Figure 4 demonstrates our algorithm for deformable collision detection and response. In the simulation, the hands are held fixed while the feet twist on the ground plane causing the legs to self-collide. To demonstrate rigid body collisions, we deform the armadillo mesh with rigid cylinders as seen in figure 2. The interactions with the cylinders demonstrate the time coherency of the strain energy local minima achieved by using the previous equilibrium state as an initial guess for the Newton-Raphson solver.

Inertia effects are neglected when simulating quasistatic elasticity, and deformation is primarily driven by external time dependent forces due to contact, collision and boundary conditions. As a result, quasistatic simulations are particularly well suited for flesh deformation where the flesh is rigidly attached to bones and heavily influenced by contact, collision and self-collision. We demonstrate the applicability of our approach with several simulations of flesh and muscles in the upper torso, derived from the visible human data set as in [TSSB\*05].

In figure 3, we attach the deformable flesh directly to the underlying skeleton. The flesh naturally deforms from the influence of the skeleton as well as from self collision, providing realistic deformation and wrinkling of the outer skin. The flesh mesh consists of 600 thousand tetrahedral elements and was simulated with a neo-Hookean constitutive model extended to the inverted regime as in [ITF04]. Figure 5 shows skeletal muscle in the upper limb simulated with the muscle constitutive model outlined in [ITF04] and [TBNF03]. Although our quasistatic formulation was only presented for isotropic materials, it is readily extended to the case of simple transverse isotropy, since the strain energy is a sum of an isotropic and a transversely isotropic component with each term being a function of their respective associated invariants. This property leads to a stiffness matrix which is a sum of an isotropic term (which can be processed in the standard fashion) and a simple anisotropic term whose eigenstructure is easy to manipulate. The resulting simulations are enriched by muscle activations that are computed from the skeletal motion as in [TSSB\*05] to produce realistic contractile motion. Finally, figure 6 shows a layered approach where we use the simulated motion of the skeletal muscles as kinematic boundary conditions for a second flesh only simulation to create more realistic muscle based skin deformation. During the second simulation, flesh nodes are constrained to follow the muscle motion if they are within a tolerance of the musculoskeletal surface.

The originally scattered armadillo geometry of figure 1 consists of 380K tetrahedra and converged to steady state in 80 Newton-Raphson iterations requiring 2-3 seconds each, under a neo-Hookean constitutive model (collision handling disabled). For the same 380K element armadillo mesh in figure 2 the computational cost was approximately 90 seconds per frame. The flesh mesh of figure 6 consisted of 600K tetrahedra and was simulated at 2 minutes per frame. All simulations were performed on a 3 GHz pentium 4 workstation. We stress that these are rather large simulation meshes, and meshes on the order of 10 thousand elements can be typically simulated at rates of 5–10 frames per second (computational cost scales nonlinearly). This is with tight bounds on the tolerance, where additional Newton-Raphson iterations lead to no visible changes. Additionally, the authors of [SNF05] use our method for a highly constrained face simulation application and report running times that translate to approximately 30 seconds per frame for a 370K tetrahedron mesh with full self and rigid body collision handling, as opposed to 50 minutes per frame, on average, for a fully dynamic simulation. Moreover, their use of quasistatic (as opposed to dynamic) simulation allows them to construct a full system Jacobian enabling the solution of an inverse problem to find muscle activations based on surface deformation.

## 12. Conclusions

We presented a framework for efficient and robust quasistatic simulation of nonlinear elastic materials using a modified Newton-Raphson algorithm that can robustly iterate through configurations that give rise to mesh inversion and buckling instabilities. Fast conjugate gradient solvers can be used, since we enforce positive definiteness of the modified linear equilibrium equations at each iteration. This simulation technique is ideal for constrained objects influenced by the motion of their specified boundary conditions. In particular, it is useful for simulating deformable flesh and skin for virtual characters whose motion is driven by an underlying kinematic skeleton.

## 13. Acknowledgements

Research supported in part by an ONR YIP award and a PECASE award (ONR N00014-01-1-0620), a Packard Foundation Fellowship, a Sloan Research Fellowship, ONR N00014-03-1-0071, ONR N00014-02-1-0720, ARO DAAD19-03-1-0331, NSF IIS-0326388, NSF ITR-0205671 and NIH U54-GM072970. E.S. was supported in part by a Stanford Graduate Fellowship, and G.I. was supported in part by a National Science Foundation Graduate Research Fellowship. We would like to thank Mike Houston, Christos Kozyrakis, Mark Horowitz, Bill Dally and Vijay Pande for computing resources.

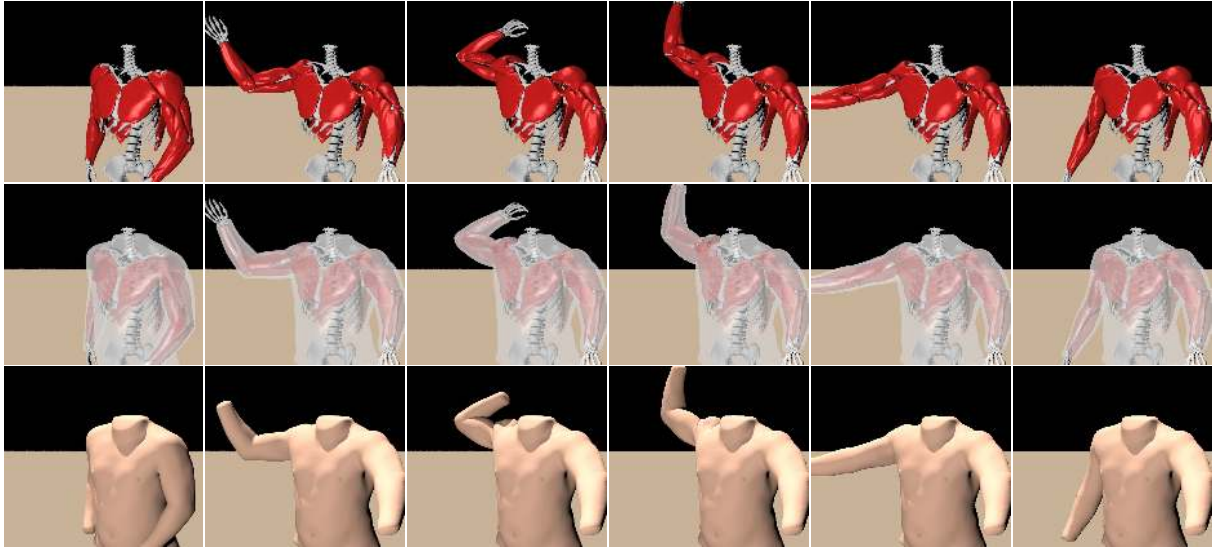
## References

- [ACP02] ALLEN B., CURLESS B., POPOVIC Z.: Articulated body deformation from range scan data. In *Proc. of ACM SIGGRAPH 2002* (2002), pp. 612–619.
- [AHS03] ALBRECHT I., HABER J., SEIDEL H. P.: Construction and animation of anatomically based human hand models. In *Proc. of the 2003 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.* (2003), pp. 98–109.
- [AS96] AMBROSIO L., SONER H. M.: Level set approach to mean curvature flow in arbitrary codimension. *J. of Differential Geometry* 43 (1996), 693–737.
- [BFA02] BRIDSON R., FEDKIW R., ANDERSON J.: Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph. (SIGGRAPH Proc.)* 21 (2002), 594–603.
- [BMF03] BRIDSON R., MARINO S., FEDKIW R.: Simulation of clothing with folds and wrinkles. In *Proc. of the 2003 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.* (2003), pp. 28–36.
- [BW98] BARAFF D., WITKIN A.: Large steps in cloth simulation. In *Proc. SIGGRAPH 98* (1998), pp. 1–12.
- [BWK03] BARAFF D., WITKIN A., KASS M.: Untangling cloth. *ACM Trans. Graph. (SIGGRAPH Proc.)* 22 (2003), 862–870.



- [CGC\*02a] CAPELL S., GREEN S., CURLESS B., DUCHAMP T., POPOVIĆ Z.: Interactive skeleton-driven dynamic deformations. *ACM Trans. Graph. (SIGGRAPH Proc.)* 21 (2002), 586–593.
- [CGC\*02b] CAPELL S., GREEN S., CURLESS B., DUCHAMP T., POPOVIĆ Z.: A multiresolution framework for dynamic deformations. In *ACM SIGGRAPH Symp. on Comput. Anim.* (2002), ACM Press, pp. 41–48.
- [CK02] CHOI K.-J., KO H.-S.: Stable but responsive cloth. *ACM Trans. Graph. (SIGGRAPH Proc.)* 21 (2002), 604–611.
- [CK05] CHOI M. G., KO H.-S.: Modal warping: Real-time simulation of large rotational deformation and manipulation. *IEEE Trans. Viz. Comput. Graph.* 11 (2005), 91–101.
- [CZ92] CHEN D., ZELTZER D.: Pump it up: Computer animation of a biomechanically based model of muscle using the finite element method. *Comput. Graph. (SIGGRAPH Proc.)* (1992), 89–98.
- [DCKY02] DONG F., CLAPWORTHY G., KROKOS M., YAO J.: An anatomy-based approach to human muscle modeling and deformation. *IEEE Trans. Vis. Comput. Graph.* 8, 2 (2002).
- [DDCB01] DEBUNNE G., DESBRUN M., CANI M., BARR A.: Dynamic real-time deformations using space & time adaptive sampling. In *Proc. SIGGRAPH 2001* (2001), vol. 20, pp. 31–36.
- [FL01] FISHER S., LIN M. C.: Deformed distance fields for simulation of non-penetrating flexible bodies. In *Comput. Anim. and Sim. '01* (2001), Proc. Eurographics Workshop, pp. 99–111.
- [GBF03] GUENDELMAN E., BRIDSON R., FEDKIW R.: Nonconvex rigid bodies with stacking. *ACM Trans. Graph. (SIGGRAPH Proc.)* 22, 3 (2003), 871–878.
- [GHDS03] GRINSPUN E., HIRANI A., DESBRUN M., SCHRODER P.: Discrete shells. In *Proc. of the 2003 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.* (2003), pp. 62–67.
- [GKS02] GRINSPUN E., KRYSL P., SCHRODER P.: CHARMS: A simple framework for adaptive simulation. *ACM Trans. Graph. (SIGGRAPH Proc.)* 21 (2002), 281–290.
- [GMTT89] GOURRET J.-P., MAGNENAT-THALMANN N., THALMANN D.: Simulation of object and human skin deformations in a grasping task. *Comput. Graph. (SIGGRAPH Proc.)* (1989), 21–30.
- [GMW81] GILL P. E., MURRAY W., WRIGHT M. H.: *Practical Optimization*. Academic Press, San Diego, USA, 1981.
- [GW03] GUILKEY J., WEISS J.: Implicit time integration for the material point method: Quantitative and algorithmics comparison with the finite element method. *Int. J. Numer. Meth. Engng* 57 (2003), 1323–1338.
- [HFS\*01] HIROTA G., FISHER S., STATE A., LEE C., FUCHS H.: An implicit finite element method for elastic solids in contact. In *Proc. of Computer Animation* (2001), pp. 136–146.
- [ITF04] IRVING G., TERAN J., FEDKIW R.: Invertible finite elements for robust simulation of large deformation. In *Proc. of the ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.* (2004), pp. 131–140.
- [JF03] JAMES D., FATAHALIAN K.: Precomputing interactive dynamic deformable scenes. *ACM Trans. Graph. (SIGGRAPH Proc.)* 22 (2003), 879–887.
- [JP02] JAMES D., PAI D.: DyRT: Dynamic response textures for real time deformation simulation with graphics hardware. *ACM Trans. Graph. (SIGGRAPH Proc.)* 21 (2002), 582–585.
- [KJP02] KRY P. G., JAMES D. L., PAI D. K.: Eigenskin: real time large deformation character skinning in hardware. In *Proceedings of the ACM SIGGRAPH symposium on Computer animation* (2002), ACM Press, pp. 153–159.
- [KM04] KURIHARA T., MIYATA N.: Modeling deformable human hands from medical images. In *Proc. of the 2004 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.* (2004), pp. 365–373.
- [KMGB04] KAUTZMAN R., MAIOLO A., GRIFFIN D., BUEKER A.: Jiggly bits and motion retargetting: Bringing the motion of Hyde to life in Van Helsing with dynamics. In *SIGGRAPH 2004 Sketches & Applications* (2004), ACM Press.
- [LCF00] LEWIS J., CORDNER M., FONG N.: Pose space deformations: A unified approach to shape interpolation a nd skeleton-driven deformation. *Comput. Graph. (SIGGRAPH Proc.)* (2000), 165–172.
- [MAC04] MARCHAL D., AUBERT F., CHAILLOU C.: Collision between deformable objects using fast-marching on tetrahedral models. In *Proceedings of the ACM SIGGRAPH symposium on Computer animation* (2004), ACM Press.
- [MBF04] MOLINO N., BAO J., FEDKIW R.: A virtual node algorithm for changing mesh topology during simulation. *ACM Trans. Graph. (SIGGRAPH Proc.)* 23 (2004), 385–392.
- [MBTF03] MOLINO N., BRIDSON R., TERAN J., FEDKIW R.: A crystalline, red green strategy for meshing highly deformable objects with tetrahedra. In *12th Int. Meshing Roundtable* (2003), pp. 103–114.
- [MDM\*02] MULLER M., DORSEY J., MCMILLAN L., JAGNOW R., CUTLER B.: Stable real-time deformations.

- In *ACM SIGGRAPH Symp. on Comput. Anim.* (2002), pp. 49–54.
- [MG03] MOHR A., GLEICHER M.: Building efficient, accurate character skins from examples. *ACM Transactions on Graphics* 22, 3 (2003), 562–568.
- [MG04] MULLER M., GROSS M.: Interactive virtual materials. In *Graph. Interface* (May 2004), pp. 239–246.
- [MKN\*04] MULLER M., KEISER R., NEALEN A., PAULY M., CROSS M., ALEXA M.: Point based animation of elastic, plastic and melting objects. In *Proc. of the 2004 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.* (2004), pp. 141–151.
- [MMDJ01] MULLER M., MCMILLAN L., DORSEY J., JAGNOW R.: Real-time simulation of deformation and fracture of stiff materials. In *Comput. Anim. and Sim. '01* (2001), Proc. Eurographics Workshop, Eurographics Assoc., pp. 99–111.
- [NTHF02] NG-THOW-HING V., FIUME E.: Application-specific muscle representations. In *Proc. of Gr. Inter. 2002* (2002), Sturzlinger W., McCool M., (Eds.), Canadian Information Processing Society, pp. 107–115.
- [OBH02] O'BRIEN J., BARGTEIL A., HODGINS J.: Graphical modeling of ductile fracture. *ACM Trans. Graph. (SIGGRAPH Proc.)* 21 (2002), 291–294.
- [OF02] OSHER S., FEDKIW R.: *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, 2002. New York, NY.
- [OH99] O'BRIEN J., HODGINS J.: Graphical modeling and animation of brittle fracture. In *Proc. SIGGRAPH 99* (1999), vol. 18, pp. 137–146.
- [PDA01] PICINBONO G., DELINGETTE H., AYACHE N.: Non-linear and anisotropic elastic soft tissue models for medical simulation. In *IEEE Int. Conf. Robot. and Automation* (2001).
- [SNF05] SIFAKIS E., NEVEROV I., FEDKIW R.: Automatic determination of facial muscle activations from sparse motion capture marker data. *to appear in ACM Trans. Graph. (SIGGRAPH Proc.)* (2005).
- [SPCM97] SCHEEPERS F., PARENT R., CARLSON W., MAY S.: Anatomy-based modeling of the human musculature. *Comput. Graph. (SIGGRAPH Proc.)* (1997), 163–172.
- [SRC01] SLOAN P., ROSE C., COHEN M.: Shape by example. In *Proc. of 2001 Symp. Int. 3D Graph.* (2001), pp. 135–143.
- [ST04] STINSON W., THURIOT P.: Bulging muscle and sliding skin: Deformation systems for Hellboy. In *SIGGRAPH 2004 Sketches & Applications* (2004), ACM Press.
- [TBNF03] TERAN J., BLEMKER S., NG V., FEDKIW R.: Finite volume methods for the simulation of skeletal muscle. In *Proc. of the 2003 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.* (2003), pp. 68–74.
- [TF88a] TERZOPOULOS D., FLEISCHER K.: Deformable models. *The Visual Computer*, 4 (1988), 306–331.
- [TF88b] TERZOPOULOS D., FLEISCHER K.: Modeling inelastic deformation: viscoelasticity, plasticity, fracture. *Comput. Graph. (SIGGRAPH Proc.)* (1988), 269–278.
- [THMG04] TESCHNER M., HEIDELBERGER B., MULLER M., GROSS M.: A versatile and robust model for geometrically complex deformable solids. In *Proc. Computer Graphics International* (2004), pp. 312–319.
- [TPBF87] TERZOPOULOS D., PLATT J., BARR A., FLEISCHER K.: Elastically deformable models. *Comput. Graph. (Proc. SIGGRAPH 87)* 21, 4 (1987), 205–214.
- [TSSB\*05] TERAN J., SIFAKIS E., SALINAS-BLEMKER S., NG-THOW-HING V., LAU C., FEDKIW R.: Creating and simulating skeletal muscle from the visible human data set. *IEEE Trans. on Vis. and Comput. Graph.* 11, 3 (2005), 317–328.
- [TW88] TERZOPOULOS D., WITKIN A.: Physically based models with rigid and deformable components. In *Graphics Interface* (1988), pp. 146–154.
- [TWS80] TAYLOR R., WILSON E., SACKET S.: Direct solution of equations by frontal and variable band active column methods. In *Europe-U.S. Workshop: Nonlinear Finite Element Analysis in Structural Mechanics* (1980), Springer-Verlag.
- [VT00] VOLINO P., THALMAN N.: Implementing fast cloth simulation with collision response. In *Proceedings of the International Conference on Computer Graphics* (2000), IEEE Computer Society, p. 257.
- [WP02] WANG X. C., PHILLIPS C.: Multi-weight enveloping: Least-squares approximation techniques for skin animation. In *Proc. ACM SIGGRAPH Symposium on Computer Animation* (2002), pp. 129–138.
- [WV97] WILHELMS J., VAN GELDER A.: Anatomically based modeling. *Comput. Graph. (SIGGRAPH Proc.)* (1997), 173–180.
- [ZCK98] ZHU Q., CHEN Y., KAUFMAN A.: Real-time biomechanically-based muscle volume deformation using FEM. *Comput. Graph. Forum* 190, 3 (1998), 275–284.



**Figure 6:** Illustration of a layered approach where the results of a quasistatic muscle simulation are subsequently used to drive a quasistatic simulation of the outer flesh.