Invertible Isotropic Hyperelasticity using SVD Gradients

Funshing Sin[1] Yufeng Zhu[1] Yongqiang Li[1] Daniel Schroeder[1,2] Jernej Barbič[1]

[1]University of Southern California [2]Carleton College

Abstract

Implicit time integration methods are commonly used in deformable object simulations to alleviate the time step restrictions. At each simulation time step, a tangent stiffness matrix is required to implicitly advance the system forward in time. We present a method to compute the tangent stiffness matrix for the invertible finite element method. The main advantage of our method is that it is able to simulate materials with strain energy functions defined in terms of principal stretches. We demonstrate the robustness of our approach using the Ogden, Saint-Venant Kirchhoff, and the rotated linear material models.

Categories and Subject Descriptors (according to ACM CCS): I.6.8 [Simulation and Modeling]: Types of Simulation—Animation, I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling

1. Introduction

Degenerated and inverted tetrahedra would normally cause numerical instability in the finite element methods. Irving et al. [ITF04] proposed the invertible finite element method (IFEM) to robustly handle element inversion. They decompose the deformation gradient of each element using the singular value decomposition (SVD), and then compute the stress tensor based on the singular values such that it gives internal forces restoring the deformations back to the rest shape. To time step the simulation, they integrate the elastic forces explicitly and damping forces implicitly. Because the integration scheme is not fully implicit, the system is only conditionally stable. Teran et al. [TSIF05] built upon IFEM and developed a method to compute the stiffness matrix allowing implicit integration. Their computation takes in materials where the strain energy $\Psi = \Psi(I, II, III)$ is written in terms of the invariants of $\mathbf{C} = \mathbf{F}^T\mathbf{F}$, where $I = tr(\mathbf{C}), II = \mathbf{C} : \mathbf{C}, III = det(\mathbf{C})$, and $\mathbf{F}$ is the deformation gradient.

Many rubbery and biological materials are simulated with material model based on principal stretches (such as Ogden) because they account well for experimentally measured nonlinear stress-strain relationships. Those models are written in terms of the three principal stretches $\lambda_1, \lambda_2, \lambda_3$ instead of the invariants I, II, III. Because λ_i are essentially the roots of a monic cubic polynomial with coefficients I, II, III (Viète formulas), derivatives of λ_i with respect to the invariants become ill-defined when two (or all three) eigenvalues get close together (such as in the rest configuration). Gao et al. [GKJD09] described a method to compute the stiffness matrix for such models by symbolically differentiating the first Piola-Kirchhoff stress with respect to the deformation gradient. However, the stiffness density function alone produces over 2500 line of C code, and different versions of both stress and stiffness density functions have to be gener-

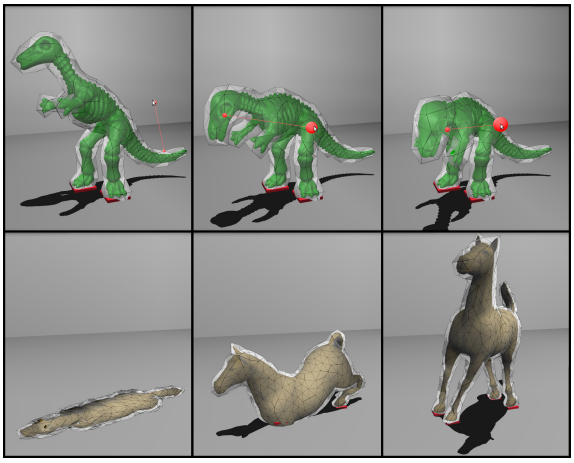

Figure 1: *A dinosaur is deformed interactively at 28 fps with $\Delta t = 1/30$ using the Ogden model. A horse collapses into a plane and recovers from severe deformation using the rotated linear model at 40 fps.*

SCA 2011, Vancouver, British Columbia, Canada, August 5 – 7, 2011.
ISBN 978-1-4503-0923-3/11/0008

ated for every possible execution path to avoid the divide-by-zero errors when λ_i get close to each other. In this poster we present an alternative method to compute the stiffness matrix that works directly on the principal stretches based material models. Our method is based on the SVD gradients computation introduced by Papadopoulo et al. [PL00].

2. Force Computation

Following Irving et al. [ITF04], we compute the force on a vertex i due to a single tetrahedron as $\mathbf{g}_i = -\mathbf{P}\mathbf{b}_i$, where $\mathbf{P} \in \mathbb{R}^{3,3}$ is the first Piola-Kirchhoff stress, $\mathbf{b}_i = (A_1\mathbf{N}_1 + A_2\mathbf{N}_2 + A_3\mathbf{N}_3)/3$, and $A_j\mathbf{N}_j$ are the area weighted material normals. We express the nodal forces in a single tetrahedron as $\mathbf{G} = \mathbf{P}\mathbf{B}_m$, where $\mathbf{G} = (\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3)$, and $\mathbf{B}_m = (\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$.

3. Stiffness Matrix Computation

The force gradient of a tetrahedron is:

$$\frac{\partial \mathbf{G}}{\partial \mathbf{u}} = \frac{\partial \mathbf{G}}{\partial \mathbf{F}}\frac{\partial \mathbf{F}}{\partial \mathbf{u}} = \left(\frac{\partial \mathbf{P}}{\partial \mathbf{F}}\mathbf{B}_m\right)\frac{\partial \mathbf{F}}{\partial \mathbf{u}} \in \mathbb{R}^{9,12} \tag{1}$$

where $\mathbf{u} \in \mathbb{R}^{12}$ is the displacement of the vertices, and $\partial\mathbf{F}/\partial\mathbf{u}$ and $\mathbf{B}_m$ are constant matrices that do not change during the simulation. The force gradient of the remaining vertex can be calculated as $\partial\mathbf{g}_0/\partial\mathbf{u} = -(\partial\mathbf{g}_1/\partial\mathbf{u} + \partial\mathbf{g}_2/\partial\mathbf{u} + \partial\mathbf{g}_3/\partial\mathbf{u})$.

Since $\mathbf{P}$ is rotationally invariant for isotropic materials, it can be computed as:

$$\mathbf{P} = \mathbf{P}(\mathbf{F}) = \mathbf{U}\mathbf{P}(\hat{\mathbf{F}})\mathbf{V}^T \tag{2}$$

where $\mathbf{F} = \mathbf{U}\hat{\mathbf{F}}\mathbf{V}^T$ (from SVD). By the product rule:

$$\frac{\partial \mathbf{P}}{\partial \mathbf{F}_{ij}} = \frac{\partial \mathbf{U}}{\partial \mathbf{F}_{ij}}\mathbf{P}(\hat{\mathbf{F}})\mathbf{V}^T + \mathbf{U}\frac{\partial \mathbf{P}(\hat{\mathbf{F}})}{\partial \mathbf{F}_{ij}}\mathbf{V}^T + \mathbf{U}\mathbf{P}(\hat{\mathbf{F}})\frac{\partial \mathbf{V}^T}{\partial \mathbf{F}_{ij}} \tag{3}$$

To compute $\partial\mathbf{U}/\partial\mathbf{F}_{ij}$, $\partial\mathbf{P}(\hat{\mathbf{F}})/\partial\mathbf{F}_{ij}$, and $\partial\mathbf{V}^T/\partial\mathbf{F}_{ij}$, we solve the following equation by taking the derivative of $\mathbf{F}$ with respect to $\mathbf{F}_{ij}$, and then multiply $\mathbf{U}^T$ and $\mathbf{V}$ on the left and right, respectively:

$$\begin{aligned}\mathbf{U}^T\left(\frac{\partial \mathbf{F}}{\partial \mathbf{F}_{ij}}\right)\mathbf{V} &= \mathbf{U}^T\left(\frac{\partial \mathbf{U}}{\partial \mathbf{F}_{ij}}\hat{\mathbf{F}}\mathbf{V}^T + \mathbf{U}\frac{\partial \hat{\mathbf{F}}}{\partial \mathbf{F}_{ij}}\mathbf{V}^T + \mathbf{U}\hat{\mathbf{F}}\frac{\partial \mathbf{V}^T}{\partial \mathbf{F}_{ij}}\right)\mathbf{V} \\ &= \left(\underbrace{\mathbf{U}^T\frac{\partial \mathbf{U}}{\partial \mathbf{F}_{ij}}}_{\tilde{\omega}_U^{ij}}\right)\hat{\mathbf{F}} + \frac{\partial \hat{\mathbf{F}}}{\partial \mathbf{F}_{ij}} + \hat{\mathbf{F}}\left(\underbrace{\frac{\partial \mathbf{V}^T}{\partial \mathbf{F}_{ij}}\mathbf{V}}_{\tilde{\omega}_{VT}^{ij}}\right)\end{aligned} \tag{4}$$

Note that $\partial\mathbf{F}_{mn}/\partial\mathbf{F}_{ij} = 0$ when $(m,n) \neq (i,j)$, and 1 otherwise. Matrices $\tilde{\omega}_U^{ij}$ and $\tilde{\omega}_{VT}^{ij}$ are antisymmetric. Since $\hat{\mathbf{F}} \in \mathbb{R}^{3,3}$ is a diagonal matrix and the diagonal elements of $\tilde{\omega}_U^{ij}\hat{\mathbf{F}}$ and $\hat{\mathbf{F}}\tilde{\omega}_{VT}^{ij}$ are zero, the three unknown entries of the diagonal matrix $\partial\hat{\mathbf{F}}/\partial\mathbf{F}_{ij}$ can be set to the diagonal of the resulting matrix on the left hand side (i.e., $diag(\mathbf{U}^T(\partial\mathbf{F}/\partial\mathbf{F}_{ij})\mathbf{V})$). Applying the chain rule gives:

$$\frac{\partial \mathbf{P}(\hat{\mathbf{F}})}{\partial \mathbf{F}_{ij}} = \frac{\partial \mathbf{P}(\hat{\mathbf{F}})}{\partial \hat{\mathbf{F}}}\frac{\partial \hat{\mathbf{F}}}{\partial \mathbf{F}_{ij}} = \sum_{d=1}^{3}\frac{\partial \mathbf{P}(\hat{\mathbf{F}})}{\partial \lambda_d}\frac{\partial \lambda_d}{\partial \mathbf{F}_{ij}} \tag{5}$$

Derivative $\partial\mathbf{P}(\hat{\mathbf{F}})/\partial\lambda_d = \partial^2\Psi/\partial\lambda_d^2$ is the Hessian of the elastic energy. It can be computed analytically for any particular material model.

The elements of $\tilde{\omega}_U^{ij}$ and $\tilde{\omega}_{VT}^{ij}$ can be computed by three 2x2 symmetric systems:

$$\begin{bmatrix}\lambda_l & \lambda_k \\ \lambda_k & \lambda_l\end{bmatrix}\begin{bmatrix}(\tilde{\omega}_U^{ij})_{kl} \\ (\tilde{\omega}_{VT}^{ij})_{kl}\end{bmatrix} = \begin{bmatrix}u_{ik}v_{jl} \\ -u_{ik}v_{jk}\end{bmatrix} \tag{6}$$

where $k,l = 1,2,3$, $diag(\hat{\mathbf{F}}) = (\lambda_1, \lambda_2, \lambda_3)$, $(.)_{kl}$ is the element (k,l) of the matrix, $u_{ik} = (U)_{ik}$, and $v_{ik} = (V)_{ik}$. Finally, we compute $\partial\mathbf{U}/\partial\mathbf{F}_{ij}$ and $\partial\mathbf{V}^T/\partial\mathbf{F}_{ij}$ as:

$$\frac{\partial \mathbf{U}}{\partial \mathbf{F}_{ij}} = \mathbf{U}\tilde{\omega}_U^{ij}, \quad \frac{\partial \mathbf{V}^T}{\partial \mathbf{F}_{ij}} = \tilde{\omega}_{VT}^{ij}\mathbf{V}^T \tag{7}$$

The system in (6) becomes ill-conditioned if $\lambda_k \approx \lambda_l$. We apply the Tikhonov regularization when the absolute difference between λ_k and $\lambda_l$is smaller than 10^{-6}.

4. Results

We tested our method on two examples in Figure 1. The dinosaur and the horse are embedded into tetrahedral meshes (1031 and 2646 elements respectively) allowing fast physics with detailed surface features. Implicit backward Euler is used in both examples.

We implemented the method presented by Teran et al. [TSIF05] and found that our method works as fast as theirs. Our method took 0.00264 sec on average to compute the stiffness matrix for the principal stretch based StVK, whereas their method took 0.00333 sec for the invariant based StVK. The timing info was measured on a six-core 3.33 GHz Intel i7 CPU with 9GB RAM.

References

[GKJD09] GAO Z., KIM T., JAMES D., DESAI J.: Semi-automated soft-tissue acquisition and modeling for surgical simulation. In *Automation Science and Engineering, 2009. CASE 2009. IEEE International Conference on* (aug. 2009), pp. 268 – 273. 1

[ITF04] IRVING G., TERAN J., FEDKIW R.: Invertible finite elements for robust simulation of large deformation. In *Proc. of the Symp. on Comp. Animation 2004* (2004), pp. 131–140. 1, 2

[PL00] PAPADOPOULO T., LOURAKIS M. I. A.: Estimating the jacobian of the singular value decomposition: Theory and applications. In *Proceedings of the 6th European Conference on Computer Vision-Part I* (2000), ECCV '00, pp. 554–570. 2

[TSIF05] TERAN J., SIFAKIS E., IRVING G., FEDKIW R.: Robust quasistatic finite elements and flesh simulation. In *2005 ACM SIGGRAPH / Eurographics Symp. on Computer Animation* (July 2005), pp. 181–190. 1, 2