

Volume Conserving Finite Element Simulations of Deformable Models

Geoffrey Irving*
Stanford University
Pixar Animation Studios

Craig Schroeder*
Stanford University

Ronald Fedkiw*
Stanford University
Industrial Light + Magic

Abstract

We propose a numerical method for modeling highly deformable nonlinear incompressible solids that conserves the volume locally near each node in a finite element mesh. Our method works with arbitrary constitutive models, is applicable to both passive and active materials (e.g. muscles), and works with simple tetrahedra without the need for multiple quadrature points or stabilization techniques. Although simple linear tetrahedra typically suffer from locking when modeling incompressible materials, our method enforces incompressibility per node (in a one-ring), and we demonstrate that it is free from locking. We correct errors in volume without introducing oscillations by treating position and velocity in separate implicit solves. Finally, we propose a novel method for treating both object contact and self-contact as linear constraints during the incompressible solve, alleviating issues in enforcing multiple possibly conflicting constraints.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling

Keywords: deformable solids, incompressibility, collisions

1 Introduction

Recently virtual humans have received increased attention for modeling stunt doubles, virtual surgery, etc. When modeling virtual humans, one needs to consider shape changes dictated by muscles, skin, fat, and other organs. These soft biological tissues are highly incompressible and involve complicated constitutive models including anisotropy and both active and passive components. Notably, volume in biological tissues is conserved locally, and it is insufficient to only conserve the total volume. Besides realistic modeling of tissues for virtual humans, volume preservation is important in its own right. [Lasseter 1987] states, “The most important rule to squash and stretch is that, no matter how squashed or stretched out a particular object gets, its volume remains constant.”

Although our interest is in physically based simulation, constant volume deformations are also of interest in shape modeling, e.g. [Yoon and Kim 2006; Angelidis et al. 2006; von Funck et al. 2006]. Several authors have proposed methods that conserve total but not local volume, e.g. [Promayon et al. 1996; Punak and Peters 2006], and [Hong et al. 2006] proposed an ad hoc method to address the “undesirable behaviors” caused by conserving only total volume.

*e-mail: {irving,cas43,fedkiw}@cs.stanford.edu

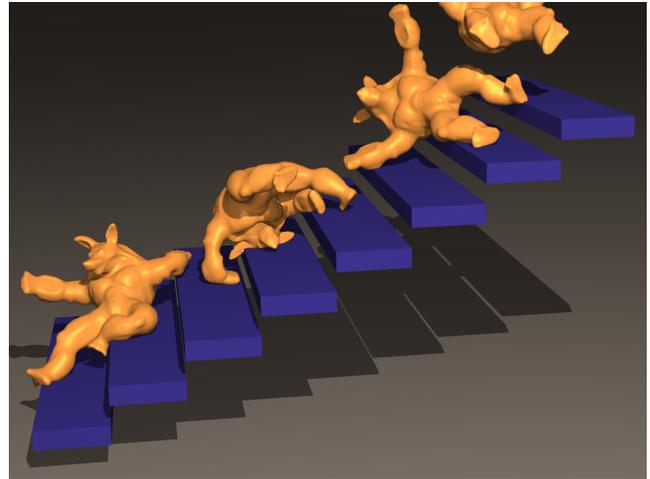


Figure 1: An incompressible elastic armadillo falls down a flight of stairs preserving its volume to an accuracy of 0.1%.

A number of authors have considered approximate local volume preservation using simple spring-like forces, e.g. [Cooper and Mad-dock 1997; Nedel and Thalmann 1998; Bourguignon and Cani 2000; Molino et al. 2003; Teschner et al. 2004]. In the area of finite element simulation, [Picinbono et al. 2001] added a volume preserving force to each tetrahedron, a technique similar to the notion of quasi-incompressibility (see [Simo and Taylor 1991]) which has been used extensively in finite element simulations of muscle tissue [Weiss et al. 1996; Teran et al. 2005]. See also [Platt and Barr 1988; Desbrun and Gascuel 1995]. The problem with mass-spring and quasi-incompressible formulations is that they only provide a force towards volume preservation, and therefore volume is not preserved in the presence of competing forces. This can be alleviated to some extent by increasing the stiffness of the volume-preserving forces, but this competes with and can overwhelm the other forces in the model.

In fluid dynamics, volume preservation is addressed by decomposing a vector field into the gradient of the pressure plus a divergence-free part and subsequently discarding the gradient (see e.g. [Fedkiw et al. 2001]). By introducing this pressure variable, one discards compressible motions while retaining those orthogonal to volume change. [Nixon and Lobb 2002] proposed a fluid dynamics approach to incompressible deformable solids, but used artificial compressibility (requiring ad hoc volume adjustments) rather than fully divergence-free velocities and did not consider constitutive models or elastic forces in the object’s interior. We take a fluid dynamics approach to deformable solids as well, introducing a pressure variable into a standard finite volume approximation. Others, such as [Roth et al. 1998], have used an independent pressure variable for similar purposes. Besides using the pressure to obtain a divergence-free velocity, we also project the positions to exactly conserve volume avoiding error accumulation (note that Eulerian fluids do not have a position variable). Similar approaches are currently receiving attention in the computational mechanics literature, see e.g. [Dolbow and Devan 2004; Oñate et al. 2004; Lahiri et al. 2005; Bijelonja et al. 2005; Bijelonja et al. 2006; Rojek et al.

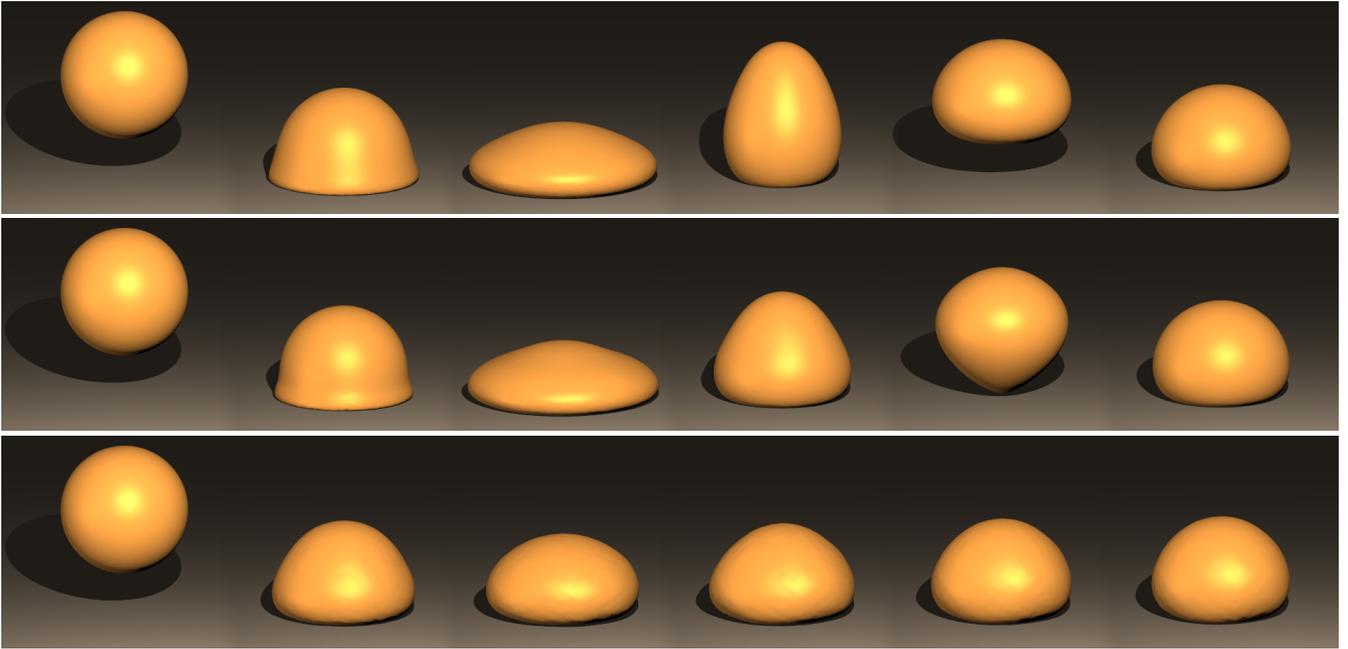


Figure 2: An elastic sphere dropped on the ground. (Top) Enforcing the volume of each one-ring using our method maintains correct volume within 1%. (Middle) Using standard finite element forces with a Poisson’s ratio of .45 results in a maximum volume loss of over 15%. (Bottom) Increasing the Poisson’s ratio to .499 reduces the maximum volume loss to 2% but causes severe locking of the sphere’s degrees of freedom hindering deformation.

2006; Cockburn et al. 2006]. In contrast to most of these works, we present a simple technique independent of any particular constitutive model or time integration scheme so that it is easily integrated into any finite element solver. Moreover, we show how to integrate this incompressibility constraint with other possibly competing constraints, in particular object contact and self-contact.

2 Time Discretization

Regardless of the time integration scheme, our goal is to make the velocity divergence free as well as to update the positions in a manner that moves the nodes to maintain constant volume in one-rings. When processing collisions or treating volume errors, some methods can only modify the position via adjustments to the velocity. Unfortunately, this requires $O(1)$ velocity changes to make $O(\Delta x)$ changes in position (since $\Delta t \sim \Delta x$). Thus, we propose a method that treats errors in position and velocity separately, ensuring that volume errors due to collisions or other phenomena can be corrected without introducing oscillations. Note that structural integration methods such as [Lahiri et al. 2005; Kharevych et al. 2006] do not have this property: in order to quickly correct $O(\Delta x)$ errors in volume they must produce $O(1)$ velocities.

For concreteness, we give the particular time integration scheme used for our examples below, with the two additional steps required for incompressibility highlighted, and then explain these new steps in detail. We used a modified version of the semi-implicit Newmark scheme of [Bridson et al. 2003] (see [Selle et al. 2007] for details). A step of size Δt from (x^n, v^n) to (x^{n+1}, v^{n+1}) proceeds as follows:

1. $v_*^{n+1/2} = v^n + \frac{\Delta t}{2} a(t^{n+1/2}, x^n, v_*^{n+1/2})$
2. $\tilde{v}^{n+1/2} = v_*^{n+1/2} + \gamma_x$ (to correct positions)
3. Modify $\tilde{v}^{n+1/2}$ with elastic and inelastic self-repulsions

4. $\tilde{x}^{n+1} = x^n + \Delta t \tilde{v}^{n+1/2}$
5. Collide with objects to obtain x^{n+1} and v_*^n
6. $\tilde{v}^n = v_*^n + \gamma_v$ (to correct velocities)
7. $v^{n+1/2} = \tilde{v}^n + \frac{\Delta t}{2} a(t^{n+1/2}, x^{n+1/2}, v^{n+1/2})$
8. $v^{n+1} = 2v^{n+1/2} - \tilde{v}^n$
9. Modify v^{n+1} for inelastic self-repulsions and friction

where $x^{n+1/2} = (x^n + x^{n+1})/2$ in step 7 is the average of the initial and final positions and $a(t, x, v)$ is the acceleration due to all forces except collisions and incompressibility. Step 1 is a backward Euler solve to obtain a velocity for use in the position update, and we adjust this velocity in step 2 so that step 4 corrects the volume in each one-ring. After colliding with kinematic objects in step 5, steps 7 and 8 advance the velocity field forward in time, and we apply a correction in step 6 to make the velocity field divergence free before time evolution.

If we temporarily ignore collisions, steps 2 and 4 combine to form $x^{n+1} = x^n + \Delta t v_*^{n+1/2} + \Delta t \gamma_x$. This formula is valid for any time integration scheme that computes x^{n+1} from x^n , by defining $v_*^{n+1/2} = (x^{n+1} - x^n)/\Delta t$. The final volumes should equal the initial volumes, i.e. $V(x^{n+1}) = V(x^0)$. Substituting for x^{n+1} and linearizing gives $V(x^n) + \Delta t \mathbf{div} v_*^{n+1/2} + \Delta t \mathbf{div} \gamma_x = V(x^0)$, where \mathbf{div} is the volume-weighted divergence (see Section 3). Similar to the typical pressure correction in fluids, we use $\tilde{v}^{n+1/2} = v_*^{n+1/2} - \nabla \hat{p}/\rho$, so $\gamma_x = -\nabla \hat{p}/\rho$ where $\hat{p} = \Delta t p$ is the scaled pressure. Thus, we have $V(x^n) + \Delta t \mathbf{div} v_*^{n+1/2} - \Delta t \mathbf{div} M^{-1} \mathbf{grad} \hat{p} = V(x^0)$ where \mathbf{grad} is the volume-weighted gradient (see Section 3) and M is the diagonal mass matrix. Rearranging into standard Poisson equation form,

$$-\mathbf{div} M^{-1} \mathbf{grad} \hat{p} = -\mathbf{div} v_*^{n+1/2} - (V(x^n) - V(x^0))/\Delta t \quad (1)$$

which can be solved for \hat{p} , and then $\gamma_k = -M^{-1} \mathbf{grad} \hat{p}$. Note that \mathbf{div} is $n \times 3n$, M is $3n \times 3n$, \mathbf{grad} is $3n \times n$, $\mathbf{v}_*^{n+1/2}$ is a $3n$ -vector, and \hat{p} , $V(x^t)$, and $V(x^0)$ are n -vectors. Finally, we stress that (1) corresponds to a single step of Newton's method applied to the equation $V(x^{n+1}) = V(x^0)$.

We correct the velocity to be divergence free in step 6, although this can be executed at any point in the algorithm since it is a static projection. Taking the divergence of step 6 and setting $\nabla \cdot \tilde{\mathbf{v}}^n = 0$ yields $0 = \nabla \cdot \mathbf{v}_*^n + \nabla \cdot \gamma_v$, where γ_v is also defined as $\gamma_v = -\nabla \hat{p} / \rho$. Similar to (1) we obtain

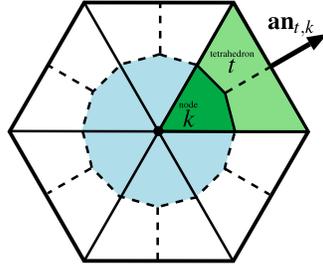
$$-\mathbf{div} M^{-1} \mathbf{grad} \hat{p} = -\mathbf{div} \mathbf{v}_*^{n+1/2} \quad (2)$$

which we can solve for \hat{p} and subsequently correct the velocity via $\tilde{\mathbf{v}}^n = \mathbf{v}_*^n - M^{-1} \mathbf{grad} \hat{p}$. The difference between (1) and (2) is that (2) computes a divergence-free velocity whereas (1) adds an extra term to obtain a non-zero divergence (similar to [Feldman et al. 2003]) in order to correct any drift in volume.

Although the velocity projection is always stable, small time steps and significant volume errors can lead to difficulties as all the missing volume is recovered at once. We alleviate this by introducing a minimum volume recovery time scale $\Delta\tau$ and clamping the last term in (1) such that its magnitude is no larger than $V(x^0)/\Delta\tau$ in any given time step.

3 Spatial Discretization

A mesh with n nodes has $3n$ degrees of freedom, and enforcing a volume constraint for each tetrahedron typically results in more than $4n$ constraints (the number of tetrahedra) making the system heavily overconstrained resulting in locking as shown in Figure 2 (bottom). We avoid locking by enforcing incompressibility on one-rings, i.e. on composite elements centered at each node, as shown in the figure to the right (the blue region). This approach adds only n constraints. Composite elements have proven useful in a number of scenarios, see e.g. [Thoutireddy et al. 2002; Boroomand and Khalilian 2004; Pires et al. 2004; de Souza Neto et al. 2005]. Note that the spatial discretization derived below is identical to the average pressure element in [Bonet and Burton 1998].



We use a standard finite volume discretization with all information collocated on the nodes of the mesh as in [Teran et al. 2003]. Let p_0 to p_3 and \mathbf{x}_0 to \mathbf{x}_3 be the pressures and positions of the four vertices of a tetrahedron. Define $P = (p_1 - p_0 \ p_2 - p_0 \ p_3 - p_0)$, $\mathbf{D} = (\mathbf{x}_1 - \mathbf{x}_0 \ \mathbf{x}_2 - \mathbf{x}_0 \ \mathbf{x}_3 - \mathbf{x}_0)$, and $\dot{\mathbf{D}} = (\mathbf{v}_1 - \mathbf{v}_0 \ \mathbf{v}_2 - \mathbf{v}_0 \ \mathbf{v}_3 - \mathbf{v}_0)$, and let V be the volume of the tetrahedron, \mathbf{an}_k the outward-facing area-weighted normal opposite vertex k , and $\mathbf{B} = \mathbf{V}\mathbf{D}^{-T} = -(\mathbf{an}_1 \ \mathbf{an}_2 \ \mathbf{an}_3)/3$. The linearly interpolated velocity field is $\mathbf{v}(\mathbf{x}) = \dot{\mathbf{D}}\mathbf{D}^{-1}(\mathbf{x} - \mathbf{x}_0) + \mathbf{v}_0$, from which $\nabla \cdot \mathbf{v}(\mathbf{x}) = \mathbf{tr}(\dot{\mathbf{D}}\mathbf{D}^{-1}) = \mathbf{D}^{-T} : \dot{\mathbf{D}}$.

The total volume-weighted divergence over the one-ring centered at node k is

$$(\mathbf{div} \mathbf{v})_k = \frac{1}{4} \int_{R(k)} \nabla \cdot \mathbf{v} dx = \frac{1}{4} \sum_{t \in R(k)} V_t \mathbf{D}_t^{-T} : \dot{\mathbf{D}}_t = \frac{1}{4} \sum_{t \in R(k)} \mathbf{B}_t : \dot{\mathbf{D}}_t$$

where $R(k)$ is the set of tetrahedra incident on k , and the fact that the

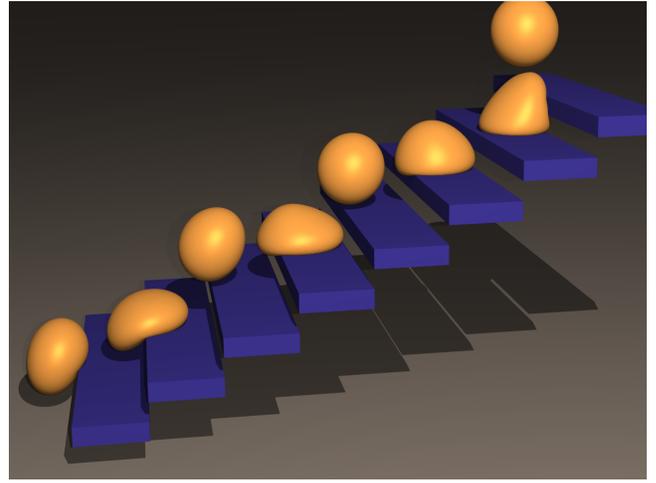


Figure 3: Incompressible elastic sphere falls down a flight of stairs illustrating rigid body collisions and contact.

divergence of the velocity is constant on each tetrahedron allows us to assign $1/4$ of the tetrahedral volume to each incident node.

We construct our gradient operator to be the negative transpose of the divergence operator so that (1) and (2) result in symmetric positive definite systems allowing for fast iterative techniques such as conjugate gradient. Thus, we want $\langle \nabla p, \mathbf{v} \rangle = \langle p, -\nabla \cdot \mathbf{v} \rangle$, that is $\int_{\Omega} \nabla p \cdot \mathbf{v} dx + \int_{\Omega} p \nabla \cdot \mathbf{v} dx = \int_{\Omega} \nabla \cdot (p\mathbf{v}) dx = \int_{\partial\Omega} p\mathbf{v} \cdot dS = 0$. This holds, for example, in the case of Dirichlet boundary conditions ($p = 0$ on the boundary). Note that we cannot define gradient using an analogue of the formula used to define divergence, since the resulting forces would not conserve momentum near the boundary.

In order to define \mathbf{grad} , we assume that the pressure field is zero outside the object (noting that it is straightforward to relax this restriction). We then partition the pressure field into $p = \sum_{t \in R(k)} p_t$ where p_t is a pressure field that agrees with p in t and is identically zero elsewhere. This allows us to restrict attention to a single tetrahedron, since the linearity of the gradient operator gives $(\mathbf{grad} p)_k = \sum_{t \in R(k)} (\mathbf{grad} p_t)_k$. Each tetrahedron in $R(k)$ makes a contribution to $(\mathbf{div} \mathbf{v})_k$ of the form $(\mathbf{div} \mathbf{v})_k = \frac{1}{4} \mathbf{B} : \dot{\mathbf{D}} = -\frac{1}{12} \sum_{j=1}^3 (\mathbf{v}_j - \mathbf{v}_0) \cdot \mathbf{an}_j = -\frac{1}{12} \sum_{j=0}^3 \mathbf{v}_j \cdot \mathbf{an}_j$. Thus, we can interpret the divergence operator on a single tetrahedron as a 4×12 matrix $-\mathbf{div} = \frac{1}{12} (\mathbf{N} \ \mathbf{N} \ \mathbf{N} \ \mathbf{N})^T$ where $\mathbf{N} = (\mathbf{an}_0^T \ \mathbf{an}_1^T \ \mathbf{an}_2^T \ \mathbf{an}_3^T)^T$.

We can now write gradient as a 12×4 matrix $\mathbf{grad} = \frac{1}{12} (\mathbf{N} \ \mathbf{N} \ \mathbf{N} \ \mathbf{N})$ so that the contribution of a single tetrahedron to the gradient at a node k is $(\mathbf{grad} p)_k = \frac{1}{12} \mathbf{N}_k \sum_{j=0}^3 p_j = \frac{1}{3} \mathbf{an}_k \bar{p}$, where \bar{p} is the average pressure of the four vertices of the tetrahedron. Summing over $R(k)$ gives

$$(\mathbf{grad} p)_k = \frac{1}{3} \sum_{t \in R(k)} \bar{p}_t \mathbf{an}_{t,k}$$

where \bar{p}_t is the average pressure in t and $\mathbf{an}_{t,k}$ is the area-weighted normal of the face opposite node k . This equation is exactly the standard FVM force for a Cauchy stress of \bar{p}_t (see [Teran et al. 2003]) and can be computed by forming $\mathbf{G} = -\mathbf{B}_t \bar{p}_t$ and distributing the columns of \mathbf{G} to the nodes (where one node gets the negation of the sum of the columns). In particular our volume preservation forces conserve momentum for each tetrahedron independent of other tetrahedra, since the net force on a tetrahedron is $\sum_k \mathbf{F}_k = -\sum_k (\mathbf{grad} p)_k = -\frac{1}{3} \bar{p} \sum_k \mathbf{an}_k = 0$. Angular

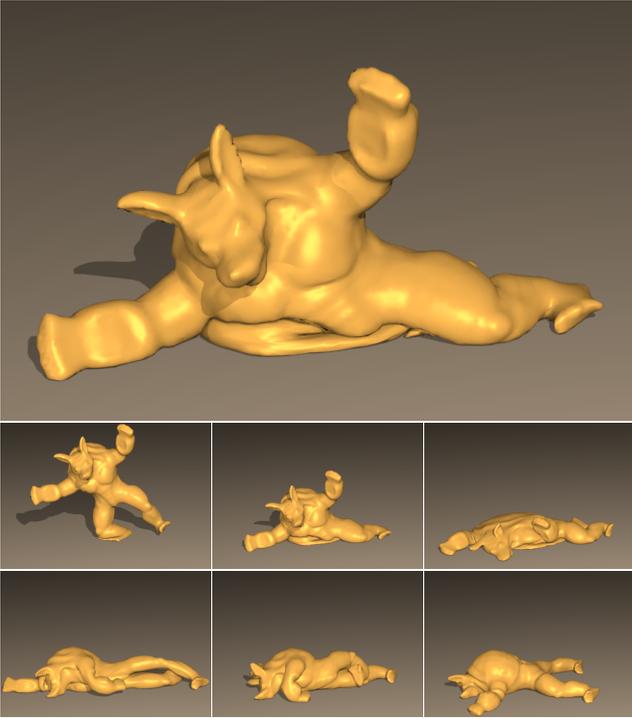


Figure 4: An incompressible elastic armadillo drops onto the ground illustrating self-collisions and contact.

momentum is also conserved per tetrahedron, since the torque is $\tau = \sum_k (\mathbf{x}_k - \mathbf{c}) \times \mathbf{F}_k = [\sum_k (\mathbf{x}_k - \mathbf{x}_0) \times \mathbf{F}_k] + [(\mathbf{x}_0 - \mathbf{c}) \times \sum_k \mathbf{F}_k]$. The second term is zero since $\sum_k \mathbf{F}_k = 0$, and replacing \mathbf{F}_k with $-(\mathbf{grad} p)_k$ makes the first term equal to $-\frac{1}{3} \bar{p} \sum_k (\mathbf{x}_k - \mathbf{x}_0) \times \mathbf{a}_k$, which is zero by Jacobi's identity.

4 Collisions and Contact

While steps 2 and 6 of the time integration algorithm work to preserve incompressibility, steps 5, 3, and 9 add additional constraints for object collisions and self-collisions. In practice, we have noticed that the blind application of our algorithm can cause serious artifacts due to these competing constraints, resulting in unusably tangled surfaces. Therefore, we incorporate both object contact and self-contact constraints into our incompressible Poisson equations. Despite being important for robust behavior in the presence of collisions, this coupling is largely undiscussed by previous authors who focused primarily on integrating incompressibility into their particular time or space discretization schemes.

Step 5 sets the position and velocity of particles to respect collisions with the object, and the conjugate gradient solver used in step 7 incorporates constraints in the normal direction to maintain the correct normal velocity for colliding particles, i.e. $\mathbf{n}^T \Delta \mathbf{v} = 0$ where \mathbf{n} is the local unit normal to the collision body and $\Delta \mathbf{v}$ is the change in velocity due to conjugate gradient. We incorporate a similar constraint into the Poisson equations solved in steps 2 and 6 of the algorithm, stressing that this is a linear constraint of the form $\mathbf{c}^T \Delta \mathbf{v} = 0$. Self-contact can similarly be written as linear constraints of the form $\mathbf{c}^T \Delta \mathbf{v} = 0$. Constraining the relative velocity of a point and triangle to not change yields

$$\mathbf{n}^T (\Delta \mathbf{v}_p - w_1 \Delta \mathbf{v}_1 - w_2 \Delta \mathbf{v}_2 - w_3 \Delta \mathbf{v}_3) = 0$$

where w_i are the barycentric weights of the point on the triangle interacting with particle p and \mathbf{n} is the triangle's normal. Constraining

the relative velocity of interacting points in an edge-edge pair yields

$$\mathbf{s}^T ((1 - \alpha_1) \Delta \mathbf{v}_1 + \alpha_1 \Delta \mathbf{v}_2 - (1 - \alpha_2) \Delta \mathbf{v}_3 - \alpha_2 \Delta \mathbf{v}_4) = 0$$

where α_i are positions of the interacting points along the segments and \mathbf{s} is the shortest vector between the interacting segments. Self-contact constraints are generated for each point-triangle and edge-edge pair currently in close proximity (these correspond to the repulsion pairs in [Bridson et al. 2002]). Note that the ability to set the velocity before the Poisson solve and guarantee no changes during the solve is equivalent to using a Neumann boundary condition on the pressure.

First consider a single constraint, i.e. a single point-object, point-triangle, or edge-edge interaction. We project out any constraint violating contribution by redefining $\gamma = -\mathbb{P}M^{-1}\mathbf{grad} \hat{p}$ altering the left-hand sides of (1) and (2) to $-\mathbf{div} \mathbb{P}M^{-1}\mathbf{grad} \hat{p}$, where \mathbb{P} projects a change in velocity using an impulse \mathbf{j} defined by $\mathbb{P}\Delta \mathbf{v} = \Delta \mathbf{v} + M^{-1}\mathbf{j}$. The impulse \mathbf{j} can be found by minimizing the kinetic energy $\mathbf{j}^T M^{-1} \mathbf{j} / 2$ subject to $\mathbf{c}^T \mathbb{P}\Delta \mathbf{v} = 0$ using the objective function $\mathbf{j}^T M^{-1} \mathbf{j} / 2 + \lambda (\mathbf{c}^T \Delta \mathbf{v} + \mathbf{c}^T M^{-1} \mathbf{j})$. Differentiating with respect to \mathbf{j} and setting to zero gives $\mathbf{j} = -\mathbf{c}\lambda$, and substituting this into the constraint equation yields $\lambda = (\mathbf{c}^T M^{-1} \mathbf{c})^{-1} \mathbf{c}^T \Delta \mathbf{v}$. Thus, $\mathbb{P}\Delta \mathbf{v} = (\mathbf{I} - M^{-1} \mathbf{c} (\mathbf{c}^T M^{-1} \mathbf{c})^{-1} \mathbf{c}^T) \Delta \mathbf{v}$ and

$$\mathbb{P} = \mathbf{I} - M^{-1} \mathbf{c} (\mathbf{c}^T M^{-1} \mathbf{c})^{-1} \mathbf{c}^T.$$

Note that $\mathbb{P}M^{-1}$ is symmetric positive semidefinite with exactly one zero eigenvalue.

In the case of many constraints $\mathbf{C}^T \Delta \mathbf{v} = (\mathbf{c}_1 \cdots \mathbf{c}_n)^T \Delta \mathbf{v} = 0$ applying the projections in simple Gauss-Seidel order gives $\mathbb{P}_n \cdots \mathbb{P}_1 M^{-1}$ which is only symmetric if none of the constraints overlap. For example, this is violated whenever point-triangle or edge-edge pairs share vertices since the corresponding \mathbb{P}_i 's do not commute. One might attempt to alleviate this problem by avoiding sequential application and applying all constraints at once, but this would require inversion of the $n \times n$ matrix $\mathbf{C}^T M^{-1} \mathbf{C}$ appearing in $\mathbb{P} = \mathbf{I} - M^{-1} \mathbf{C} (\mathbf{C}^T M^{-1} \mathbf{C})^{-1} \mathbf{C}^T$ which is prohibitively expensive for complex scenarios with dynamic constraints. Instead, we apply the projections in alternating forward and backward Gauss-Seidel sweeps using the symmetric positive semidefinite matrix

$$\mathbb{P}M^{-1} = (\mathbb{P}_1 \cdots \mathbb{P}_n \cdots \mathbb{P}_1)^q M^{-1}$$

where q is a small integer. Applying a single unsatisfied projection \mathbb{P}_i strictly reduces the energy, so this iteration is stable and always converges to the correct constraint satisfying velocity. In practice, we found that using only $q = 4$ iterations reduced the constraint violating components by 1-2 orders of magnitude on average. Since the pressure system is itself solved to only 1% accuracy, and the alternating sweeps ensure symmetry of the full matrix regardless of convergence, this was sufficient for all our examples. These projections increase the cost of solving for pressure only moderately since there are typically many fewer collisions than vertices and each \mathbb{P}_i can be applied in constant time. Since object and point-triangle contacts have more coherent normals and are typically better behaved than edge-edge contacts, we bias unconverged results towards the former by placing them first (and last) in the ordering.

The final matrix $-\mathbf{div} \mathbb{P}M^{-1}\mathbf{grad} \hat{p}$ is always symmetric positive semidefinite, but can become singular in cases with large numbers of constraints. Since conjugate gradient breaks down for singular matrices, we instead use MINRES, an alternative Krylov space method which requires only symmetry of the matrix. MINRES required significantly fewer iterations than conjugate gradient even in nonsingular cases (though this could change with preconditioning). For example, Figure 2 averaged 72 iterations with conjugate gradient and 34 iterations with MINRES (with small additional cost

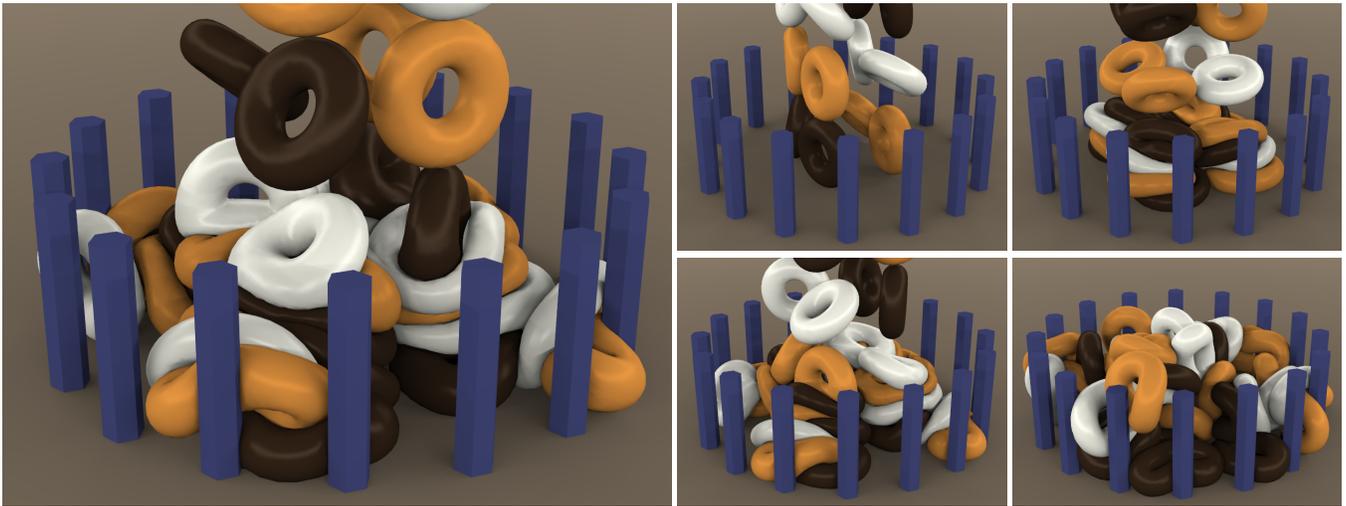


Figure 5: 40 incompressible elastic tori fall into a pile illustrating complex collision and contact. Object contact and self-contact are represented as linear constraints during each Poisson solve. Each torus maintains correct volume to within 0.5%, even at the bottom of the pile.

per iteration). Thorough description and analysis of MINRES, conjugate gradient, and related solvers for singular or nearly singular systems can be found in [Choi 2006].

5 Examples

We used the method of [Irving et al. 2004] for internal deviatoric finite element forces in all our examples. When necessary, we used a minimum volume recovery time scale of one-fifth of a frame. Figure 2 shows a comparison of our method against a standard finite volume discretization using a 104k element mesh. Using a 3GHz Xeon machine, the computational cost was 18 s/frame for our method, 25 s/frame with Poisson’s ratio .45, and 3.4 min/frame with Poisson’s ratio .499. Similarly the simulation time for Figure 3 was 34 s/frame. The armadillo simulations in Figures 1 and 4 were both under 4 min/frame with a 112k element mesh. The simulation in Figure 5 took an average of 15 min/frame for 40 12k element meshes (500k elements total) with approximately 65% the time spent in the two Poisson solves due to the complexity of the contact constraints.

As a stress test for our method, we squeezed an incompressible sphere with 22k elements between two kinematic plates (similar to an example in [Hong et al. 2006]). The minimum volume recovery time scale was not used (i.e. we set $\tau = 0$). The sphere was successfully compressed to 1.1% of its original thickness before numerical error forced the time step to zero (all computations were performed

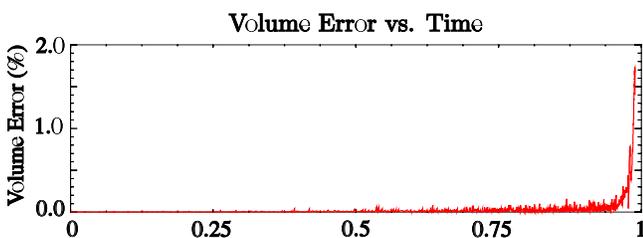


Figure 6: Volume error, in percent, as a sphere is pressed between plates. The plates are just touching the sphere at time 0 and move towards each other with constant velocity until they meet at time 1.

in single precision). The total volume error remained below 1.7% throughout the simulation, and was lower than 0.1%, 0.5%, and 1% until the sphere reached 13%, 2.3%, and 1.4% of its original thickness, respectively. A plot of volume error vs. time is shown in Figure 6. For this simulation we modified the time integration scheme to enforce contact constraints during both backward Euler solves (steps 1 and 7) instead of only in step 7, so that the volume correction in step 2 used the correct collision-aware velocities for particles in contact with the plates. The use of uncorrected velocities as input to volume correction would have caused significant degradation for this simulation due to the high tension involved. This modification was not necessary for the other examples, which is fortunate since enforcing contact constraints in both solves typically causes sticking artifacts during separation.

6 Conclusion

We proposed a novel technique for enforcing local incompressibility in deformable solids drawing ideas from computational fluid dynamics. We benefit from the simplicity and flexibility of tetrahedra while avoiding the pitfalls of locking by enforcing volume preservation over one-rings instead of individual tetrahedra. We augmented our method to incorporate both object contact and self-contact constraints into the incompressible solve to alleviate problems with conflicting constraints. The method is trivially adapted for triangles and thin shells.

Acknowledgments

Research supported in part by an ONR YIP award and a PECASE award (ONR N00014-01-1-0620), a Packard Foundation Fellowship, ONR N0014-06-1-0393, ONR N00014-05-1-0479 for a computing cluster, ARO DAAD19-03-1-0331, NIH U54-GM072970, NSF CCF-0541148, NSF IIS-0326388 and NSF ITR-0205671. G.I. was supported in part by an NSF Graduate Fellowship.

References

ANGELIDIS, A., CANI, M., WYVILL, G., AND KING, S. 2006. Swirling-sweepers: constant volume modeling. *Graphical Models* 68, 4, 324–32.

- BIJELONJA, I., DEMIRDŽIĆ, I., AND MUZAFERIJA, S. 2005. A finite volume method for large strain analysis of incompressible hyperelastic materials. *Int. J. Numer. Meth. Eng.* 64, 1594–1609.
- BIJELONJA, I., DEMIRDŽIĆ, I., AND MUZAFERIJA, S. 2006. A finite volume method for incompressible linear elasticity. *Comput. Methods Appl. Meth. Eng.* 195, 6378–90.
- BONET, J., AND BURTON, A. 1998. A simple average nodal pressure tetrahedral element for incompressible and nearly incompressible dynamic explicit applications. *Comm. Num. Meth. Eng.* 14, 437–449.
- BOROOMAND, B., AND KHALILIAN, B. 2004. On using linear elements in incompressible plane strain problems: a simple edge based approach for triangles. *Int. J. Num. Meth. Eng.* 61, 1710–1740.
- BOURGUIGNON, D., AND CANI, M. P. 2000. Controlling anisotropy in mass-spring systems. In *Eurographics*, Eurographics Assoc., 113–123.
- BRIDSON, R., FEDKIW, R., AND ANDERSON, J. 2002. Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph. (SIGGRAPH Proc.)* 21, 594–603.
- BRIDSON, R., MARINO, S., AND FEDKIW, R. 2003. Simulation of clothing with folds and wrinkles. In *Proc. of the 2003 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 28–36.
- CHOI, S.-C. 2006. *Iterative Methods for Singular Linear Equations and Least-Squares Problems*. PhD thesis, Stanford Univ.
- COCKBURN, B., SCHÖTZAU, D., AND WANG, J. 2006. Discontinuous Galerkin methods for incompressible elastic materials. *Comput. Methods Appl. Meth. Eng.* 195, 3184–3204.
- COOPER, L., AND MADDOCK, S. 1997. Preventing collapse within mass-spring-damper models of deformable objects. In *5th Int. Conf. in Central Europe on Comput. Graph. and Vis.*
- DE SOUZA NETO, E., PIRES, F. A., AND OWEN, D. 2005. F-bar-based linear triangles and tetrahedra for finite strain analysis of nearly incompressible solids. part i: formulation and benchmarking. *Int. J. Num. Meth. Eng.* 62, 353–383.
- DESBRUN, M., AND GASCUEL, M.-P. 1995. Animating soft substances with implicit surfaces. In *Proc. SIGGRAPH 95*, 287–290.
- DOLBOW, J., AND DEVAN, A. 2004. Enrichment of enhanced assumed strain approximation for representing strong discontinuities: Addressing volumetric incompressibility and the discontinuous patch test. *Int. J. Numer. Meth. Eng.* 59, 47–67.
- FEDKIW, R., STAM, J., AND JENSEN, H. 2001. Visual simulation of smoke. In *Proc. of ACM SIGGRAPH 2001*, 15–22.
- FELDMAN, B. E., O'BRIEN, J. F., AND ARIKAN, O. 2003. Animating suspended particle explosions. *ACM Trans. Graph. (SIGGRAPH Proc.)* 22, 3, 708–715.
- HONG, M., JUNG, S., CHOI, M., AND WELCH, S. 2006. Fast volume preservation for a mass-spring system. *IEEE Comput. Graph. and Appl.* 26 (September), 83–91.
- IRVING, G., TERAN, J., AND FEDKIW, R. 2004. Invertible finite elements for robust simulation of large deformation. In *Proc. of the ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 131–140.
- KHAREVYCH, L., YANG, W., TONG, Y., KANSO, E., MARSDEN, J., SCHRÖDER, P., AND DESBRUN, M. 2006. Geometric variational integrators for computer animation. *ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 43–51.
- LAHIRI, S., BONET, J., PERAIRE, J., AND CASALS, L. 2005. A variationally consistent fractional time-step integration method for incompressibility and nearly incompressible lagrangian dynamics. *Int. J. Numer. Meth. Eng.* 59, 1371–95.
- LASSETER, R. 1987. Principles of traditional animation applied to 3D computer animation. *Comput. Graph. (SIGGRAPH Proc.)*, 35–44.
- MOLINO, N., BRIDSON, R., TERAN, J., AND FEDKIW, R. 2003. A crystalline, red green strategy for meshing highly deformable objects with tetrahedra. In *12th Int. Meshing Roundtable*, 103–114.
- NEDEL, L. P., AND THALMANN, D. 1998. Real time muscle deformations using mass-spring systems. In *Proc. Comput. Graph. Int.*, 156–165.
- NIXON, D., AND LOBB, R. 2002. A fluid-based soft-object model. *IEEE Comput. Graph. Appl.* 22, 4, 68–75.
- OÑATE, E., ROJEK, J., TAYLOR, R., AND ZIENKIEWICZ, O. 2004. Finite calculus formulation for incompressible solids using linear triangles and tetrahedra. *Int. J. Numer. Meth. Eng.* 59, 1473–1500.
- PICINBONO, G., DELINGETTE, H., AND AYACHE, N. 2001. Non-linear and anisotropic elastic soft tissue models for medical simulation. In *IEEE Int. Conf. Robot. and Automation*.
- PIRES, F. A., DE SOUZA NETO, E., AND OWEN, D. 2004. On the finite element prediction of damage growth and fracture initiation in finitely deforming ductile materials. *Comput. Meth. in Appl. Mech. and Eng.* 193, 5223–5256.
- PLATT, J. C., AND BARR, A. H. 1988. Constraint methods for flexible models. *Comput. Graph. (SIGGRAPH Proc.)*, 279–288.
- PROMAYON, E., BACONNIER, P., AND PUECH, C. 1996. Physically-based deformations constrained in displacements and volume. In *Eurographics*, vol. 15, Eurographics Assoc.
- PUNAK, S., AND PETERS, J. 2006. Localized volume preservation for simulation and animation. In *Poster, SIGGRAPH Proc.*, ACM.
- ROJEK, J., NATE, E. O., AND TAYLOR, R. 2006. Cbs-based stabilization in explicit solid dynamics. *Int. J. Numer. Meth. Eng.* 66, 1547–68.
- ROTH, S. H., GROSS, M., TURELLO, M. H., AND CARLS, S. 1998. A Bernstein-Bézier based approach to soft tissue simulation. *Comput. Graph. Forum (Proc. Eurographics)* 17, 3, 285–294.
- SELLE, A., SU, J., IRVING, G., AND FEDKIW, R. 2007. Highly detailed folds and wrinkles for cloth simulation. In *Proc. of ACM SIGGRAPH/Eurographics Symp. on Comput. Anim. (in review)*.
- SIMO, J., AND TAYLOR, R. 1991. Quasi-incompressible finite elasticity in principal stretches: continuum basis and numerical examples. *Comput. Meth. in Appl. Mech. and Eng.* 51, 273–310.
- TERAN, J., BLEMKER, S., NG, V., AND FEDKIW, R. 2003. Finite volume methods for the simulation of skeletal muscle. In *Proc. of the 2003 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 68–74.
- TERAN, J., SIFAKIS, E., SALINAS-BLEMKER, S., NG-THOW-HING, V., LAU, C., AND FEDKIW, R. 2005. Creating and simulating skeletal muscle from the visible human data set. *IEEE Trans. on Vis. and Comput. Graph.* 11, 3, 317–328.
- TESCHNER, M., HEIDELBERGER, B., MÜLLER, M., AND GROSS, M. 2004. A versatile and robust model for geometrically complex deformable solids. In *Proc. Comput. Graph. Int.*, 312–319.
- THOUTIREDDY, P., MOLINARI, J., REPETTO, E., AND ORTIZ, M. 2002. Tetrahedral composite finite elements. *Int. J. Num. Meth. Eng.* 53, 1337–1351.
- VON FUNCK, W., THEISEL, H., AND SEIDEL, H.-P. 2006. Vector field based shape deformations. *ACM Trans. Graph. (SIGGRAPH Proc.)* 25, 3, 1118–1125.
- WEISS, J., MAKER, B., AND GOVINDJEE, S. 1996. Finite-element implementation of incompressible, transversely isotropic hyperelasticity. *Comput. Meth. in Appl. Mech. and Eng.* 135, 107–128.
- YOON, S., AND KIM, M. 2006. Sweep-based freeform deformations. In *Eurographics*, vol. 25, Eurographics Assoc., 487–96.