Animating Developable Surfaces using Nonconforming Elements

Elliot English[*]
University of British Columbia

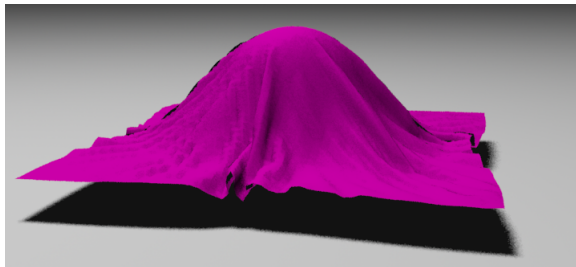Robert Bridson[†]
University of British Columbia

Figure 1: *Frames from an animation of developable cloth, with complex collisions.*

Abstract

We present a new discretization for the physics-based animation of developable surfaces. Constrained to not deform at all in-plane but free to bend out-of-plane, these are an excellent approximation for many materials, including most cloth, paper, and stiffer materials. Unfortunately the conforming (geometrically continuous) discretizations used in graphics break down in this limit. Our nonconforming approach solves this problem, allowing us to simulate surfaces with zero in-plane deformation as a hard constraint. However, it produces discontinuous meshes, so we further couple this with a "ghost" conforming mesh for collision processing and rendering. We also propose a new second order accurate constrained mechanics time integration method that greatly reduces the numerical damping present in the usual first order methods used in graphics, for virtually no extra cost and sometimes significant speed-up.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: cloth, developable surface, finite elements, constraints

1 Introduction

Many deformable surfaces, ranging from most types of cloth to paper [Kergosien et al. 1994; Bo and Wang 2007] and stiffer materials, are well approximated as *developable*: they bend out-of-plane but do not visibly stretch or compress in-plane. Even for materials which do allow some in-plane deformation, e.g. small amounts of shearing in a fabric relative to the warp and weft directions, if the simulator cannot handle the developable limit there are bound to be

---
[*]e-mail: eenglish@cs.ubc.ca
[†]e-mail: rbridson@cs.ubc.ca

numerical problems as users attempt to approach it. We thus restrict our attention in this paper to the fully developable case, imposing zero in-plane deformation as a hard constraint, though of course our technique is easily generalized to stretchy or shearable materials.

Unfortunately, standard graphics simulators break down precisely at this limit. For example, for a triangle mesh with the usual piecewise linear elements, developability implies that each triangle remain rigid. For any nontrivial bending this constraint must be violated: the mesh can essentially only crease along straight lines already present in the mesh as edges: it *locks*.

The general phenomena of locking, i.e. the inability of a given finite element space to approximate solutions [Brenner and Scott 2002; Hauth 2004], was recently brought to light in graphics for volume-conserving volumetric simulations by Irving et al. [2007]. In the developable surface case, Liu et al.'s rigorous analysis shows that a general n-triangle conforming mesh only has $O(\sqrt{n})$ degrees of freedom [2007], with significant mesh-dependent artifacts. Similar arguments show that quad meshes with bilinear elements suffer from the same locking problem, as do many higher order polynomial elements. For stiff but not fully constrained cloth models, e.g. where edges may change their length slightly, locking manifests as a spurious increased resistance to bending, proportional to the in-plane stiffness rather than the true bending stiffness.
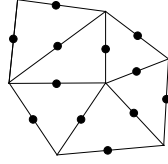
The classic solution to the locking problem for volume-conserving deformations is to use *nonconforming* elements [Brenner and Scott 2002]. Rather than reduce the number of constraints by averaging over larger regions as Irving et al. propose, finite element practitioners traditionally increase the number of variables, by putting the variables at the midpoints of edges (in 2D) or faces (in 3D).[1] We adopt this approach for the deformable surface case, putting our position variables at the midpoints of the edges rather than the vertices in the mesh (see figure 2). This now gives us $3e \approx 9v$ variables, and to make each linear triangle rigid only implies $3t \approx 6v$ constraints, leaving us with approximately $3v$ true degrees of freedom for bending—allowing the method to accurately approximate developable surfaces. The first part of our paper gives the details on this approach: how to enforce developability, special treatment of boundary elements, and a simple bending model.

However, the discrete surfaces we work with are nonconforming, i.e. no longer necessarily continuous: adjacent triangles only have to meet at the midpoint of the common edge, not necessarily at vertices. This clearly poses a problem for robust collision processing

---
[1]This may be identified as the lowest order Crouzeix-Raviart element.

Figure 2: *Schematic of nonconforming variables, located at midpoints of edges between triangles. While continuous at these points, the surface may be discontinuous along the rest of each edge.*

and rendering. We therefore couple a "ghost" conforming mesh (with the usual vertex variables) to the simulation, used just for collisions and rendering.

Finally, since we impose developability as a hard constraint, we turn to time integration of constrained mechanics. The usual schemes in graphics unfortunately suffer from strong numerical damping with nonlinear constraints, as energy is erroneously transferred to constrained modes and projected out. We propose a new second order accurate multistep method, based on BDF2 and simple position-based constraint projection. This both reduces numerical damping and speeds projection (since we remain closer to the constraint manifold), without need for stabilization or velocity projection.

2 Previous Work

Cloth simulation has a long history within computer graphics; we highlight here just a sampling of the relevant papers from the standpoint of the developable limit.

Provot [1995] worked with mass-spring models, introducing a loose constraint on edges to not deform by more than 10% with a simple Gauss-Seidel iteration. Bridson et al. [2003] demonstrated improved buckling behavior if edges were (loosely) constrained to not compress at all, just stretch. The critical aspect of this approach, though not identified at the time, was that triangles were left with some freedom to deform, fortuitously avoiding the locking problem. Of course, this can't realistically handle the many materials which more severely limit strain, and the Gauss-Seidel constraint iteration tends to induce mesh-dependent artifacts. This paper also makes use of the robust collision processing algorithm (for conforming triangle meshes) developed by Bridson et al. [2002] from earlier work by Provot [1997].

Baraff and Witkin [1998] instead proposed a semi-implicit Backwards Euler integrator to avoid the stability time step restriction plaguing explicit time integration of stiff models. Later authors argued that the strong numerical damping present in Backwards Euler was responsible for the fairly smooth appearance of Baraff and Witkin's results; we suspect a large share of the problem was locking, as very stiff in-plane forces brought the material model close to developable, causing spurious numerical resistance to bending.

Choi and Ko [2002] introduced the second order accurate BDF2 method to cloth simulation, which features much reduced damping yet still has stiff decay [Ascher and Petzold 1998], of crucial importance for dealing with stiff systems.[2] They also fortuitously avoided locking with their implicit model of the buckling instability, allowing edges in their model to compress easily while still offering stiff resistance to stretch. (This can also be viewed as using biphasic springs, with lower resistance to compression than stretch.) This

---

[2]By contrast, implicit symplectic integrators, such as certain Newmark schemes, cannot possess stiff decay and thus exhibit objectionable temporal aliasing of high frequency modes into low frequency modes when using large time steps.

is an attractive solution for much of the large-scale motion of the cloth, but the buckling model causes small-scale details to remain implicit—i.e. not visible in the simulation mesh. Attempts at procedurally adding in the missing detail have met with mixed success (e.g. [Volino and Magnenat-Thalmann 1999; Kang and Cho 2002; Tsiknis 2006]).

More recently Goldenthal et al. [2007] demonstrated an effective approach to constraining a quad-dominant cloth mesh to zero deformation along the warp and weft directions. They avoid locking by requiring most of the mesh to use quad elements, and by not constraining shearing: this leaves enough degrees of freedom to accurately and beautifully capture many fabrics of interest, but it cannot be extended to the developable no-shear limit or to triangle meshes. Our new constrained mechanics time integration scheme is a multistep extension of Goldenthal et al.'s fast projection method.

Bergou et al. [2006] introduced the nonconforming elements we use in the context of deriving a compact stencil for bending forces on conforming meshes; our work generalizes this to use them for in-plane dynamics to solve locking.

Liu et al. [2007] imposed developability as a constraint on conforming triangle meshes, proving that n triangles give you $O(\sqrt{n})$ degrees of freedom. Unfortunately, those degrees of freedom suffer from mesh-dependent artifacts which do not vanish under refinement: this is the clearest illustration in the literature of the locking problem we face, and motivates why we need a new method.

3 A Nonconforming Element Discretization

We begin with a regular triangle mesh in parameter or "object" space, with the midpoint of each edge i at parameter space position p_i as in figure 2. Each edge variable also has a world space position x_i, a velocity v_i, etc. Within a triangle with edges i, j, and k, we extend variables with linear interpolation/extrapolation: e.g. from geometric similarity the world space position of the vertex located at the corner opposite edge i is $x_j + x_k - x_i$. We can also phrase this in terms of piecewise linear basis functions $\{\phi_i\}$, where $\phi_i(p)$ is 1 all along edge i and zero at the midpoints of all other edges: $x(p) = \sum_i x_i \phi_i(p)$.

The mass m_i associated with edge i is simply a third of the mass of the sum of the masses of the incident triangles; these can be assembled into a diagonal mass matrix M, with each mass repeated three times. Newton's law is then $d^2x/dt^2 = M^{-1}F$, where F is a vector of the net forces on each edge.

For a regular elastic material, we could use the usual Galerkin finite element discretization (see Brenner and Scott [2002] for example), integrating gradients of the nonconforming basis functions over each triangle as appropriate to get a stiffness matrix, but—crucially—avoiding integrating over the jump discontinuities on the edges between triangles.

However in the developable limit we take, an equivalent but simpler formulation is possible. The deformation gradient in each triangle is the gradient of world space position w.r.t. parameter values. For linear elements, this is constant in each triangle; to avoid in-plane deformation, this gradient matrix must be orthogonal, i.e. the world space pose of each triangle must be a rigid transformation of the parameter space pose. A triangle is rigid if and only if the distance between any two edge midpoints remains constant, giving three constraints per triangle of the form

$$c_{ij}(x) = \|x_i - x_j\|^2 - d_{ij}^2 = 0 \qquad (1)$$

where d_{ij} is the parameter space distance between edge midpoints i and j. We assemble all of these constraints in one column-vector-valued function $C(x)$.

Figure 3: *On the left is a square of developable surface pinned at two corners: we enforce zero in-plane deformation to a relative error tolerance of 10^{-4}, giving virtually no sag along the top edge. Extra constraints ensure realistic behavior at the boundaries. To the right is a irregularly meshed nonplanar shirt worn by a moving character.*

This is now a discrete constrained mechanics problem, with Lagrange multiplier constraint forces of the form

$$F_c = \left(\frac{\partial C}{\partial x}\right)^T \lambda = J^T \lambda, \tag{2}$$

where $J = \partial C / \partial x$ is the Jacobian of the constraint function and λ is a vector containing one Lagrange multiplier per constraint. In section 5 we will discuss methods for integrating this motion.

3.1 Boundary Constraints

While this model is robust when the motion of the triangles along the edges of the mesh is prescribed, for more typical free boundary situations artifacts do arise: the per-triangle rigidity constraints aren't quite enough here. For example, a corner triangle with two boundary edges—and thus only one edge shared with another triangles—is free to rotate arbitrarily, independent of the orientations of nearby triangles. Even if a triangle only has one boundary edge, and is connected to the rest of the mesh at two edge midpoints, it is also free to spin around one axis independently of the orientations of nearby triangles. In fact, even when restricted to planar motion the model fails: the only planar developable motions are globally rigid, yet this model allows additional deformations. In the planar elasticity context, this is a known instability due to free traction boundary conditions (as opposed to displacement boundary conditions, i.e. prescribed positions): see e.g. Falk [1991].

We thus need further boundary constraints: we require that for each boundary vertex the corresponding vertices on the incident nonconforming elements match up. As a result these elements are also C^0 along their shared edges, have their orientations properly coupled, and no longer have spurious rotational freedom: the outer "ribbon" of triangles, i.e. all those that touch the boundary, are forced to be conforming while the interior of the mesh is still free to be nonconforming. Figure 3, showing a rectangle pinned at two corners, illustrates the robustness of this approach. This also resolves the planar deformation problem, since the only solution for the conforming outer ribbon is globally rigid motion, which induces a rigid position boundary condition on the interior nonconforming elements which in turn has been proven to only have the expected rigid motion as a solution [Brenner and Scott 2002].

3.2 Bending Forces

While discretization of bending forces is not the focus of this paper, obviously conforming mesh methods based on vertex unknowns can't be directly applied. However, Wardetzky et al. [2007] presented a discretization of bending forces on conforming triangle meshes which uses our nonconforming elements as an intermediary. We thus use this model, without the projection to conforming meshes, to arrive at a consistent discretization on the nonconforming mesh.

4 Collisions and Rendering

The nonconforming surface may have jump discontinuities along edges, apart from at the midpoints. Rendering this surface directly shows undesirable cracks between triangles. Similarly if the cloth collision algorithm were run on it potentially fatal scenarios could arise as the exposed edges of the nonconforming elements would allow the elements to tangle with or pass through one another.

Our solution to these problems is to maintain a ghost conforming mesh (with vertex-based positions, see figure 4) of the same topology as the nonconforming mesh, initialized with the same geometry. We begin a time step by computing candidate new edge positions x_n and velocities v_n for the nonconforming mesh based on non-collision forces (gravity, internal constraints, bending, etc.). We then transfer these to candidate new vertex positions x_c and velocities v_c for the conforming mesh, by taking the average at each vertex of the nonconforming values extrapolated from all incident triangles. We represent this with an averaging matrix A:

$$x_c = A x_n. \tag{3}$$

These give candidate conforming mesh trajectories, from the positions at the end of the last time step x_c^{old} (which are guaranteed to be non-interpenetrating) to the new conforming positions x_c, with new velocities $v_c = A v_n$. We feed this into a standard cloth collision code [Bridson et al. 2002] to solve for the final conforming positions x_c' and velocities v_c', which should be non-interpenetrating. This new intersection-free conforming mesh is saved for later rendering and collision processing in the following step.

Once operations on the conforming mesh are complete we update the nonconforming mesh to its final positions x_n', coupling the effect of collisions back into the main simulation. We assume a Lagrange multiplier form for the nonconforming correction:

$$x_n' = x_n + A^T \lambda \tag{4}$$

where λ is chosen so that averaging the final nonconforming positions x_n' back to the conforming mesh vertices returns the final conforming positions:

$$A x_n' = x_c'. \tag{5}$$

This gives a simple symmetric positive definite linear system to solve, which simplifies to:
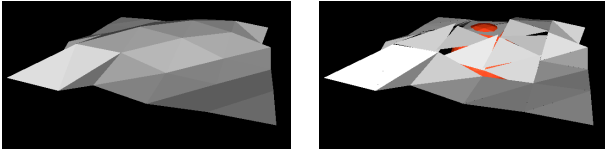
$$A A^T \lambda = x_c' - x_c. \tag{6}$$

Note that the coefficient matrix $A A^T$ is constant, thus we can run Cholesky factorization once and then very efficiently solve this at every subsequent time step. Also note that if no collisions occur, we do not modify the nonconforming positions at all, as one would expect. We finish by updating the nonconforming velocities in a consistent manner:

$$v_n' = v_n + \frac{x_n' - x_n}{\Delta t} \tag{7}$$

also noting here that in the absence of collisions we do not modify the velocities at all.

In summary, after advancing the nonconforming mesh, we update the conforming mesh by averaging vertices, run collision handling,

Figure 4: *The ghost mesh (left) closely tracks the nonconforming mesh (right) while maintaining a penetration free state with itself and the orange obstacle.*

and then use Lagrange multipliers to update the nonconforming mesh. Figures 1 and 5 show the robustness of this scheme, guaranteeing an intersection-free conforming mesh that closely tracks the simulation even in complex self-folding scenarios.

While this method has produced reasonable results, we highlight some unresolved issues. The first is that simply averaging to get the conforming mesh often doesn't provide as smooth a surface as might be desired, as might be expected since the sampling density of the conforming mesh is in fact lower. The rendered conforming mesh is also only approximately developable, of course, though its motion does reflect the developable dynamics of the underlying nonconforming mesh. We expect edge subdivision schemes (e.g. [Peters and Reif 1997]) may help with both these issues. Finally, collisions are handled after internal dynamics, introducing an $O(\Delta t)$ splitting error which can perturb developability. We have not noticed any obvious artifacts stemming from these issues, but in principle they could be a concern especially in severe collision scenarios with large time steps.

5   Time Integration of Constraints

Goldenthal et al. [2007] introduced "fast projection", a particularly attractive first order accurate time integrator for constrained mechanics. This may be derived, heuristically at least, as the limit of Backwards Euler applied to a system where hard constraints $C(x) = 0$ are replaced with very stiff elastic forces $F_c(x) = -kJ^T(x)C(x)$, the gradient of the potential $\frac{1}{2}k\|C(x)\|^2$. Backwards Euler to get to time step $n+1$ is:

$$x^{(n+1)} = x^{(n)} + \Delta t v^{(n+1)},$$
$$v^{(n+1)} = v^{(n)} + \Delta t M^{-1}(F_a - kJ^TC), \tag{8}$$

where F_a are the applied non-constraint forces (gravity and bending), and J and C are evaluated at $x^{(n+1)}$. Eliminating velocity:

$$x^{(n+1)} + k\Delta t^2 M^{-1}J^TC = x^{(n)} + \Delta t v^{(n)} + \Delta t^2 M^{-1}F_a. \tag{9}$$

Taking the limit as $k \to \infty$ can be shown, at least for constraints close enough to linear in a local neighborhood, to give the fast projection algorithm:

$$x_0 = x^{(n)} + \Delta t v^{(n)} + \Delta t^2 M^{-1}F_a$$
$$x^{(n+1)} = \text{project}(x_0). \tag{10}$$

Here x_0 is a predicted position, and the second step projects this onto the constraint manifold, so that $C(x^{(n+1)}) = 0$. The projection is implemented as a sequence of Newton-like steps, where from the current estimated position x_j the linearized constraint $C(x_j) + J(x_j)(x_{j+1} - x_j) = 0$ is solved, with a Lagrange multiplier update $x_{j+1} = x_j + J(x_j)^T\lambda$. This is terminated when $\|C(x)\|$ is below some tolerance, and can be made more robust with the usual strategies such as line search. Once the new $x^{(n+1)}$

is found, the new velocity can be determined from the Backwards Euler step $x^{(n+1)} = x^{(n)} + \Delta t v^{(n+1)}$.

While elegant and effective, this algorithm can suffer from numerical damping when the constraints are nonlinear. From a velocity perspective, even if the time n velocity is tangent to the constraint manifold at $x^{(n)}$, the tangent space changes at $x^{(n+1)}$ so the velocity components now normal to the constraint are projected out, for a net loss of kinetic energy.

We therefore propose a multistep version of this algorithm, based instead on the second order accurate multistep method BDF2, which similarly has stiff decay [1998] and thus also allows a stiff limit. Assuming constant step sizes, BDF2 can be written as:

$$x^{(n+1)} = \tfrac{4}{3}x^{(n)} - \tfrac{1}{3}x^{(n-1)} + \tfrac{2}{3}\Delta t v^{(n+1)}$$
$$v^{(n+1)} = \tfrac{4}{3}v^{(n)} - \tfrac{1}{3}v^{(n-1)} + \tfrac{2}{3}\Delta t M^{-1}F(x^{(n+1)}). \tag{11}$$

Eliminating velocity as before gives:

$$x^{(n+1)} = \tfrac{4}{3}x^{(n)} - \tfrac{1}{3}x^{(n-1)} + \tfrac{8}{9}\Delta t v^{(n)} - \tfrac{2}{9}v^{(n-1)}$$
$$+ \tfrac{4}{9}\Delta t^2 M^{-1}F(x^{(n+1)}). \tag{12}$$

We again break this up into a prediction step followed by a projection to the constraint:

$$x_0 = \tfrac{4}{3}x^{(n)} - \tfrac{1}{3}x^{(n-1)} + \tfrac{8}{9}\Delta t v^{(n)} - \tfrac{2}{9}v^{(n-1)}$$
$$+ \tfrac{4}{9}\Delta t^2 M^{-1}F \tag{13}$$
$$x^{(n+1)} = \text{project}(x_0).$$

Finally the time $n+1$ velocity is taken from the BDF2 formula for the position update, which can be rewritten as:

$$v^{(n+1)} = \tfrac{1}{\Delta t}\left[\tfrac{3}{2}x^{(n+1)} - 2x^{(n)} + \tfrac{1}{2}x^{(n-1)}\right]. \tag{14}$$
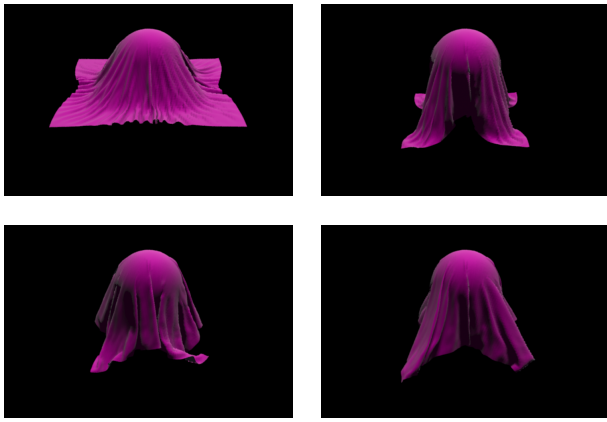
The basic steps of the algorithm are the same as before, just with a few extra vector adds in finding the predicted position and final velocity, which is of negligible cost compared to the projection operation. The error analysis is nontrivial, since at first glance the second order truncation error in the projection would seem to give a globally first order method. However, while the predicted position is indeed $O(\Delta t^2)$ away from the constraint manifold, most of this error is normal to manifold itself and is eliminated by projection: our numerical experiments indicate that after projection the position only suffers $O(\Delta t^3)$ truncation error, and together with the third order error in velocity, this results in a globally second order accurate algorithm. This can easily be verified for the simple case of motion constrained to a circle, for example: a step of length L off the circle on a tangent gives a predicted position that is $O(L^2)$ away from the circle, but its projection back to the circle is at arc-length $L + O(L^3)$ along the circle. Meanwhile, the higher order formulas for velocity prevent the rapid dissipation of energy present in the first order scheme.

Additional benefits include an order of magnitude less numerical dissipation, due to higher accuracy, and typically improved projection times: by using additional information from previous steps about the constraint manifold, the predicted position tends to stay closer to the manifold and requires fewer iterations in projection.

6   Results

Figures 1, 3 and 5 show example frames from simulations. We rendered the raw conforming mesh, with smoothed vertex normals but no subdivision. The meshes were 100×100, and simulations

Figure 5: *A developable surface is dropped on a sphere, with immediate wrinkling and creasing patterns.*

ran at 9.52 seconds/step with time steps of 1ms on an Athlon 64 3500+, with PARDISO [Schenk and Gärtner 2006] as the linear solver. We ran fast projection with a tolerance on maximum relative error of 10^{-4}, taking 10 Newton steps on average. Due to step size, collision handling was a small fraction of total run time.

We also simulated a moving skinned character wearing a cape to evaluate the new integration scheme; simulation time was reduced to 3.96 from 5.37 seconds/step for the single step scheme, as the multistep method required on average half the number of iterations for the projection stage to converge. Similarly a skinned character wearing an irregularly meshed shirt, of higher resolution than the cape, ran at 4.58 seconds/step using the new scheme.

7 Conclusions

We have presented an effective new discretization for deformable surfaces which can robustly handle developable surfaces without locking artifacts at any mesh resolution. Since the underlying nonconforming simulation mesh isn't continuous, we couple in a conforming ghost mesh to handle contact and collisions and for rendering. In addition, we provided a second order accurate multistep constrained mechanics time integration scheme based on BDF2, using just position projection, which both accelerates fast projection and significantly reduces numerical damping.

However, for stretchy materials where regular simulators work, the constant factor overhead of this method will probably make it uncompetitive. We also are particularly interested in resolving the issues brought up by the ghost conforming mesh. Edge subdivision schemes (in lieu of our simple vertex averaging) may improve the smoothness of the rendered output, and coupling internal dynamics and collisions, similar to Baraff and Witkin's method [1998], may avoid perturbations from developability caused by time splitting.

Acknowledgements

References

ASCHER, U. M., AND PETZOLD, L. R. 1998. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations.* SIAM, Philadelphia, PA, USA.

BARAFF, D., AND WITKIN, A. 1998. Large steps in cloth simulation. In *Proc. ACM SIGGRAPH*, 43–54.

BERGOU, M., WARDETZKY, M., HARMON, D., ZORIN, D., AND GRINSPUN, E. 2006. A quadratic bending model for inextensible surfaces. In *Symp. Geometry Processing*, 227–230.

BO, P., AND WANG, W. 2007. Geodesic-controlled developable surfaces for modeling paper bending. *CGF 26*, 3, 365–374.

BRENNER, S. C., AND SCOTT, L. R. 2002. *The Mathematical Theory of Finite Element Methods (2nd ed.).* Springer.

BRIDSON, R., FEDKIW, R., AND ANDERSON, J. 2002. Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph. (Proc. SIGGRAPH) 21*, 3, 594–603.

BRIDSON, R., MARINO, S., AND FEDKIW, R. 2003. Simulation of clothing with folds and wrinkles. In *Symp. Comp. Anim.*, 28–36.

CHOI, K.-J., AND KO, H.-S. 2002. Stable but responsive cloth. *ACM Trans. Graph. (Proc. SIGGRAPH) 21*, 3, 604–611.

FALK, R. S. 1991. Nonconforming finite element methods for the equations of linear elasticity. *Math. Comp. 57*, 196, 529–550.

GOLDENTHAL, R., HARMON, D., FATTAL, R., BERCOVIER, M., AND GRINSPUN, E. 2007. Efficient simulation of inextensible cloth. *ACM Trans. Graph. (Proc. SIGGRAPH) 26*, 3, 49.

HAUTH, M. 2004. *Visual Simulation of Deformable Models.* PhD thesis.

IRVING, G., SCHROEDER, C., AND FEDKIW, R. 2007. Volume conserving finite element simulations of deformable models. *ACM Trans. Graph. (Proc. SIGGRAPH) 26*, 3, 13.

KANG, Y. M., AND CHO, H. G. 2002. Bilayered approximate integration for rapid and plausible animation of virtual cloth with realistic wrinkles. In *Computer Animation, IEEE Computer Society*, 203–214.

KERGOSIEN, Y. L., GOTODA, H., AND KUNII, T. L. 1994. Bending and creasing virtual paper. *IEEE Comput. Graph. Appl. 14*, 1, 40–48.

LIU, Y.-J., TANG, K., AND JONEJA, A. 2007. Modeling dynamic developable meshes by the hamilton principle. *Comput. Aided Des. 39*, 9, 719–731.

PETERS, J., AND REIF, U. 1997. The simplest subdivision scheme for smoothing polyhedra. *ACM Trans. Gr. 16*, 4, 420–431.

PROVOT, X. 1995. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *GI*, 147–154.

PROVOT, X. 1997. Collision and self-collision handling in cloth model dedicated to design garment. *GI*, 177–89.

SCHENK, O., AND GÄRTNER, K., 2006. On fast factorization pivoting methods for symmetric indefinite systems.

TSIKNIS, K. D. 2006. *Better cloth through unbiased strain limiting and physics-aware subdivision.* Master's thesis, University of British Columbia.

VOLINO, P., AND MAGNENAT-THALMANN, N. 1999. Fast geometric wrinkles on animated surfaces. In *7th Intl. Conf. in Central Europe on Computer Graphics and Visualization (WSCG)*.

WARDETZKY, M., BERGOU, M., HARMON, D., ZORIN, D., AND GRINSPUN, E. 2007. Discrete quadratic curvature energies. *Computer Aided Geometric Design*, 499–518.