Embedded Thin Shells for Wrinkle Simulation

Olivier Rémillard
School of Computer Science
McGill University

Paul G. Kry
School of Computer Science
McGill University

Abstract

We present a new technique for simulating high resolution surface wrinkling deformations of composite objects consisting of a soft interior and a harder skin. We combine high resolution thin shells with coarse finite element lattices and define frequency based constraints that allow the formation of wrinkles with properties matching those predicted by the physical parameters of the composite object. Our two-way coupled model produces the expected wrinkling behavior without the computational expense of a large number of volumetric elements to model deformations under the surface. We use C^1 quadratic shape functions for the interior deformations, allowing very coarse resolutions to model the overall global deformation efficiently, while avoiding visual artifacts of wrinkling at discretization boundaries. We demonstrate that our model produces wrinkle wavelengths that match both theoretical predictions and high resolution volumetric simulations. We also show example applications in simulating wrinkles on passive objects, such as furniture, and for wrinkles on faces in character animation.

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

Keywords: finite elements, shells, shape functions, constraints

Links: DL PDF

1 Introduction

Wrinkles are important visual details that appear on the surface of deformable objects. If an object has a thin surface layer with elastic properties that are stiffer than those of the underlying volume, then wrinkles will form when the object is under compression. This is because the skin, being thin and stiff, will more easily undulate in order to accommodate the compression. This occurs naturally with human skin where the epidermis and dermis layers have varied thickness and elastic properties which are different from the subcutaneous tissue, fascia, and underlying muscles [Danielson 1973]. The phenomenon is also observed in other examples of composite materials at various scales, from dried fruit to mountain formation [Genzer and Groenewold 2006].

In computer graphics, many different approaches are used to model and animate wrinkling. There has been a lot of research focusing on physically based simulation of thin shells and clothing. Much of this work is relevant, and the wrinkling is similar; however, our focus is on the situation where the skin is physically attached to its foundation while clothing is typically draped or worn with intermittent collisions and contact providing the driving forces. In other work focusing on interactive solutions, procedural models have been proposed for creating wrinkles, for instance, within skinning techniques. Wrinkle maps are likewise a common approach for procedurally adding visual details to faces, clothes, or to show muscle activations. When the wrinkles always form in exactly the same places, these techniques perform well. But there are many situations where it is preferable to have a simulation, for example, in wrinkling due to arbitrary contacts with the environment.

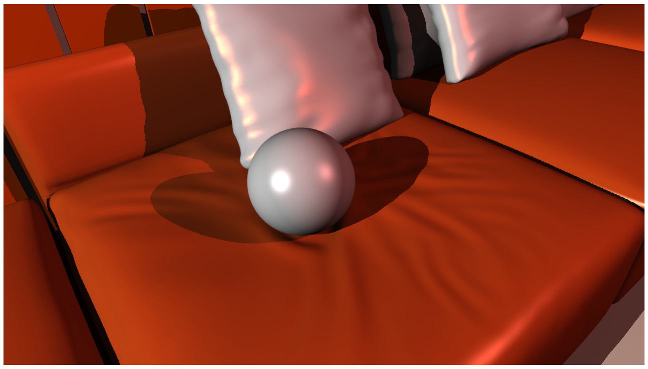

Figure 1: *An example of wrinkles produced by our technique.*

In the context of embedded meshes, the high resolution details of a surface mesh are visible, but the low number of degrees of freedom in the mechanical lattice prevents any new wrinkles from forming. While biomechanical simulations of faces and physically based skinning techniques have pushed the sheer number of elements to incredible numbers, these methods still fall short of the necessary amount of degrees of freedom to produce fine wrinkling.

Our technique builds upon the embedded mesh approach, recognizing that bulk deformations have much lower spatial frequencies in situations involving wrinkling. Our model replaces the embedded mesh with a thin shell and unites both systems with position constraints. By designing the constraints to act smoothly only at lower frequencies, we ensure that they do not interfere with wrinkle formation or large deformation. We identify the frequency cut-off using a model of wrinkle wavelength that takes as parameters the skin thickness and the elastic properties of both the skin and the interior. Furthermore, we use quadratic shape functions to model interior deformation, which allows the number of interior elements that we need to be kept to a minimum while avoiding C^0 artifacts that linear shape functions produce at element boundaries. Whereas the quadratic shape functions associate more degrees of freedom to each element, it is still straightforward to build a dynamic model lattice with superimposed elements and node duplication that properly take into account the underlying mesh connectivity. Our solver produces static solutions for the shell (which is thin and light and will not typically exhibit visual dynamics), and we let the shell deformations contribute forcing on the dynamics of the interior. Figure 1 shows a preview of the results.

2 Related Work

Early seminal work on elastic deformable models introduced techniques based on finite elements to computer animation [Terzopoulos et al. 1987; Terzopoulos and Witkin 1988]. In other early work, free form deformation lattices [Sederberg and Parry 1986] were used to define specific deformation modes, for which physically based equations of motion could be created to produce animation [Faloutsos et al. 1997]. The combination of coarse finite element models with fine meshes has become a popular computer animation technique, and is typically referred to as *embedded meshes*. Various examples of applications and modifications to the basic formulation include skeleton driven deformations [Capell et al. 2002], corotational formulation for fast simulation with linear elasticity [Müller et al. 2002], fracture [Müller and Gross 2004; Parker and O'Brien 2009], node duplication to deal with small pieces during fracture [Molino et al. 2004], and elements robust to inversion [Irving et al. 2004]. Wojtan and Turk [2008] present embedded meshes in the context of viscoelastic flow, and can produce fine folds over time due to periodic re-embedding of the plastic deformations and refinement of large or skinny triangles. Embedded meshes have also been used in numerical coarsening of inhomogeneous elastic materials [Nesme et al. 2009]. The popularity of embedded meshes in computer animation is probably due to the ease with which a model can be embedded into tetrahedra or hexahedra, in contrast to the complexity of producing exact volumetric meshing of a surface mesh. Even though meshing software is available for solving this problem, we often do not want to simulate our objects at such fine resolutions, and instead prefer an inexpensive simulation of a low dimensional model; this is also our preference in our embedded thin shell approach.

For faces, there has been previous success in generating creases and furrows using physically based models [Terzopoulos and Waters 1990; Sifakis et al. 2005]. However, these techniques have difficulty in producing fine details due to the sheer number of elements that are required to model wrinkling at the skin. Thalmann et al. [2002] specifically look at skin wrinkling properties in relation to mechanical properties in high resolution models of the dermis and epidermis in 2D cross-section simulations. With growing computer power and fast algorithms, very large numbers of elements can be simulated [McAdams et al. 2011]. This and other work related to higher resolution simulation are relevant because of our interest in simulating very fine resolution surface features. However, even with these volumetric simulation methods, we believe it is still difficult to simulate the number of elements necessary to produce fine surface wrinkles.

The simulation of wrinkles and folds has more commonly been the domain of cloth and thin shell animation. For cloth, Bridson et al. [2002] use subdivision to smooth wrinkling during the robust treatment of collisions (we similarly use subdivision in rendering some of our final results). At higher resolutions, Bridson et al. [2003] use a variety of techniques to promote the development of wrinkling details in regions where there is contact. Thin shells are much like cloth but have non flat rest configuration. We use the bending energy described by Grinspun et al. [2003], though alternatives exist, for instance a quadratic energy model for inextensible surfaces [Bergou et al. 2006]. We have explored different options for stretching and shearing energy, but we ultimately use the model of Baraff and Witkin [1998] in our thin shell simulation. Most work in graphics and animation makes use of linear shape functions due to the ease with which they can be implemented. Among the exceptions is work on cloth [Thomaszewski et al. 2006] and physically based shape editing [Mezger et al. 2009]. We use continuous B-spline quadratic shape functions for volume elements so as not to influence the location of wrinkles on our embedded shell. Although we use linear shape functions for the shell simulation, we use subdivision to smooth the result.

There has been a lot of focus on using coarse simulations and augmenting the results with fine details [Bergou et al. 2007; Kavan et al. 2011]. This is relevant to our work because of the necessity of describing constraints that link a coarse simulation of the global deformations, with a fine resolution simulation of the surface. We explore both global and local constraints and ultimately prefer local constraints, which makes our work most similar to that of Bergou et al. [2007]. An important difference is that we build our constraints based on the wavelength of wrinkles as predicted by the skin thickness and elastic properties of the model. Furthermore, we incorporate our constraints within a two-way coupled simulation that also evolves the dynamic interior of the object. We note that other interesting aspects of coupling physical models are explored by Sifakis et al. [2007]. For single point contact, Seiler et al. [2012] show how to augment a coarse simulation with fine details during user interaction. Our model, in contrast, allows for multiple points of contact as we do not precompute wrinkle patterns.

Wrinkles can be created procedurally based on stretch tensors [Rohmer et al. 2010], or even designed as part of a character skin [Larboulette and Cani 2004; Wang et al. 2006]. Other work looks specifically at wrinkles and folds of bending fingers [Venkataraman et al. 2005]. Certainly, artist designed wrinkle maps (or normal maps) are likely the most common technique for modeling and rendering fine details since the early days [Blinn 1978], and are still widely used in current applications [Jimenez et al. 2011]. In contrast to artist designed wrinkle maps or procedural models, data capture or precomputation can be used as a source of detailed surface geometry or maps. In the context of character animation, we can see our embedded thin shell technique as an interesting means for automating the creation of wrinkle map examples.

3 Coupling Surface and Interior

We start with a surface mesh and build a coarse volumetric finite element model to simulate large deformations of the interior. Depending on the context, the interior of the object could also be referred to as the foundation, or substructure, or substrate. We largely follow the standard approach used in embedded mesh simulation techniques. While there are many options, we use B-spline quadratic shape functions and a hexahedral lattice, and we describe the specifics of our implementation with respect to model construction and simulation in Sections 5 and 6.

We build a linear embedding relation $B : \mathbb{R}^m \to \mathbb{R}^n$ using interpolation weights of the surface mesh with respect to the hexahedral lattice given the mesh position at the reference configuration. The reference configuration represents the equilibrium state of the object, and can have existing geometric folds and creases as features. Given the interior reference position $q_0 \in \mathbb{R}^m$, the surface reference position $x_{q_0} \in \mathbb{R}^n$ is given by

$$x_{q_0} = Bq_0.$$

Interpolated mesh positions Bq are often sufficient to produce realistic animation featuring interesting deformations. However, fine features such as wrinkles, pinching, furrows, and grooves on the embedded surface can only be observed when the elements of the underlying lattice are sufficiently fine, requiring expensive models with hundreds of thousands, if not millions of elements. Our formulation instead addresses the problem by coupling a high resolution thin shell to the embedded mesh driven by a low resolution lattice.

To maintain the coherence between the surface and the interior, the shell must be attached to the embedded mesh. The most naive option would be to add penalty forces between the shell and the embedded mesh. This would work and would be fast, but the stiffness of the penalty will directly influence the formulation of wrinkles, both the frequency and magnitude, which complicates the selection of parameters. Instead, we couple the surface to the interior through position constraints, formulated as a matrix H providing a linear transformation $H : \mathbb{R}^n \rightarrow \mathbb{R}^c$, with $c \ll n$, which is applied to both the shell and the embedded mesh,

$$Hx = HBq.$$

The constraints must force the shell x to match the embedded shape Bq, but only at low spatial frequencies, with the null space of H allowing for the high spatial frequency deformations necessary to produce wrinkles. We discuss the expected wrinkle wavelength in Section 4, while the next two subsections present different options for defining the constraints.

3.1 Constraints with Local Support

We create constraints with local support in a manner inspired by the work of Bergou et al. [2007] on tracking thin shells. Each of our constraints requires that the weighted average of a cluster of shell vertices matches a corresponding weighted average of embedded mesh vertices. Through a careful selection of clusters and weights we can ensure that we only constrain the shell at spatial frequencies below that of the natural wrinkles (see Section 4). For a given cluster of vertices C, the constraint has the form

$$\sum_{i \in C} \alpha_i x_i = \sum_{i \in C} \alpha_i [Bq]_i \ , \tag{1}$$

where the weights α_i provide an affine combination of the shell vertex positions x_i and the embedded mesh vertex positions $[Bq]_i$. We use a truncated Gaussian function for the weights, that is,

$$\alpha_i = w_C \, e^{-\frac{d_i^2}{2\sigma^2}} \ \text{ if } \ d_i < 2\sigma, \ \ 0 \ \text{ otherwise} \ , \tag{2}$$

where w_C normalizes the sum of weights to one, and d_i is the distance from vertex i to the cluster's center, which we approximate by the shortest path following mesh edges. The standard deviation σ determines the frequency attenuation power of this Gaussian as a spatial filter. The truncation at 2σ determines vertex membership in the cluster; it is an arbitrary and conservative choice that ensures the sparsity and smoothness of the constraints.

We use a set of c overlapping clusters centered at locations distributed evenly across the surface. The cluster centers partition the mesh into regions as seen in Figure 2. Centers are greedily selected in a preprocessing step using breadth first search along mesh edges. We start by choosing a random vertex for the first center, and then compute the minimum edge traversal distance to all other vertices. We choose the farthest vertex as the next center, and then update the minimum distances for all the vertices that are closer to this center. We repeat until all vertices have at most a distance of $\frac{1}{2}r$ to a center, where r is the desired distance between centers of adjacent regions, and is a parameter determined by the material properties of the object as described in Section 4.

Each cluster defines a constraint, and together they form the rows of our sparse constraint matrix H. Typically we observe that each constraint involves a few hundred vertices, with each vertex influenced by 3 to 10 constraints. Although the normalization w_C of each cluster's weights is arbitrary in the context of the constraint, it helps our iterative solver by ensuring that clusters are treated equally despite how few or how many vertices they contain. Overall, the Gaussian weights act as a local low-pass filter, and the overlapping clusters that make up our constraints can be viewed as a filtered geometry reconstruction of limited frequency.

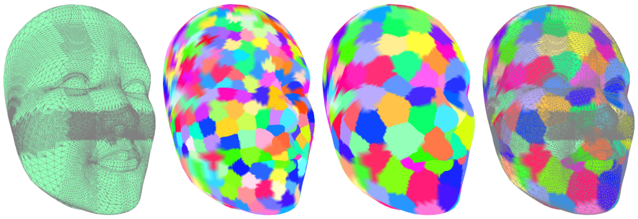

Figure 2: *An example mesh with two different resolution partitions into uniform regions. Our greedy algorithm is unaffected by varying density in the mesh. Region centers define the locations for overlapping Gaussian weighted averages that make up our constraints.*

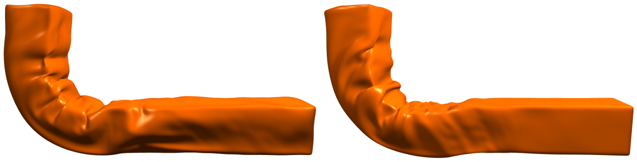

Figure 3: *When bending a rectangular solid, spectral constraints produce ripples across the entire model (left), while local constraints allow wrinkling effects to remain localized (right).*

3.2 Global Spectral Constraints

A natural choice for constraining the shell to the embedding at low spatial frequencies is to construct H based on the spectral properties of the mesh. In particular, inspired by work by Kavan et al. [2011] on tracking for cloth simulation, we can construct the rows of H from the c eigenvectors corresponding to the smallest eigenvalues of the decomposition of the mesh Laplacian. When the rows of H contain only low spectral frequencies of the mesh, the constraints leave the higher frequencies undisturbed. This approach generates the smoothest low frequency basis with the disadvantage that H is dense. However, an advantage is that it is possible to select the frequency cut off by choosing the appropriate size of the basis c from the eigenvalues.

While the global approach can also have advantages in speed for solvers that exploit orthogonality in H, we prefer constraints with local support. With local constraints, the wrinkles remain localized to where there is deformation, while global constraints tend to distribute ripples over the entire surface, as shown in Figure 3.

4 Wrinkle parameters

We base the construction of our constraints on the expected frequency of wrinkling on the surface. We use a simple model that describes the mechanical behavior of a skin attached to a soft elastic foundation [Timoshenko and Gere 2009]. When a skin of small thickness is strongly adhered to an infinitely thick substrate, buckling or wrinkling of the skin occurs only when a compressive force exceeds a critical value, at which point wrinkles form in the skin with a sinusoidal deflection profile of wavelength λ along the compression direction. Given the thickness of the skin h, and the Young's modulus and the Poisson ratio of the skin and the interior (E_x, ν_x, and E_q, ν_q respectively), the corresponding critical wavelength is

$$\lambda = 2\pi h \left[\frac{(1-\nu_q^2)\mathsf{E}_x}{3(1-\nu_x^2)\mathsf{E}_q} \right]^{1/3} . \tag{3}$$

Using the critical wrinkling wavelength for our given object's parameters, we can compute the inter-cluster spacing r and the standard deviation σ to use in the construction of our constraints. Recall that the averaging weights in a given cluster come from a Gaussian with the shape

$$g(d) = e^{-\frac{d_i^2}{2\sigma^2}} . \tag{4}$$

The parameter σ must be chosen such that this Gaussian filter attenuates spatial frequencies at and above the critical wrinkle wavelength. The Fourier transform gives us the frequency response profile of our averaging weights as

$$G(\xi) = \sigma\sqrt{\frac{\pi}{2}}e^{-2\sigma^2\pi^2\xi^2} , \tag{5}$$

which is a scaled Gaussian with standard deviation $(2\sigma\pi)^{-1}$. To assure a sufficient attenuation at the critical wavelength, we use two standard deviations and we set

$$\frac{1}{\lambda} = 2(2\sigma\pi)^{-1} , \quad \text{thus,} \quad \sigma = \lambda/\pi . \tag{6}$$

The inter cluster spacing r is also tied to wrinkle wavelength, and the Nyquist-Shannon sampling theorem dictates that $1/r$ should be less than twice the critical wrinkle frequency $1/\lambda$, otherwise our constraints can prevent the formation of wrinkles. Therefore, $r > \frac{1}{2}\lambda = \frac{1}{2}\sigma\pi$. In practice, we choose to use an inter cluster spacing of $r = 2\sigma$.

While the critical wavelength given by Equation 3 is constant, the amplitude of the buckles is related to the compression of the interior. Beyond the linear region where this simple model is accurate, the wavelength is still proportional to Equation 3.

We note that there are alternatives to our greedy approach to building well-spaced cluster centers for local constraints, and examples are discussed by Bergou et al. [2007]. Nevertheless, we observe that our simple greedy construction in Section 3.1 works well in practice. More importantly, if an object has varying elasticity and skin thickness, then we should vary the density of cluster centers and the Gaussian standard deviations of each cluster in order to accommodate the variations in wrinkle wavelength; we leave this for future work.

4.1 Selecting Parameters

There are a variety of options for setting the parameters to create physically based wrinkles. The most straightforward is to directly use the elastic properties and surface thickness parameters of the real world materials corresponding to our model. For instance, for a leather couch with foam interior, we can use a leather of thickness $h = 2$ mm and Young's modulus $\mathsf{E}_x = 40$ MPa, and foam interior with $\mathsf{E}_q = 0.5$ MPa, giving wrinkles of wavelength approximately 4 cm. The stiffness parameters for the elastic stretching and bending energy of the surface are set as $h\mathsf{E}_x$ and $h^3\mathsf{E}_x$ respectively to conform to the model parameters. We note that the Possion ratio does not play a very important part in determining the wavelength and can largely be ignored, even when the surface and interior have varying Poisson ratios (it will not change λ by more than 10%). It is the skin thickness and the Young's modulus ratio that have the most significance in determining wrinkle wavelength. If instead we want to choose parameters based on a desired visual wrinkle appearance, then we can easily choose a skin thickness and Young's modulus ratio that together produce the desired wavelength.

5 Model Composition

We use dense triangle meshes with shell stretch and shear energy defined by Baraff and Witkin [1998], and bending energy defined by Grinspun et al. [2003]. Thin shell energies are best suited to regular meshes, and have poor behavior in meshes that contain sliver triangles. In the case of problematic meshes with poorly shaped triangles we have used L_p centroidal Voronoi tessellation to remesh the surface [Lévy and Liu 2010].

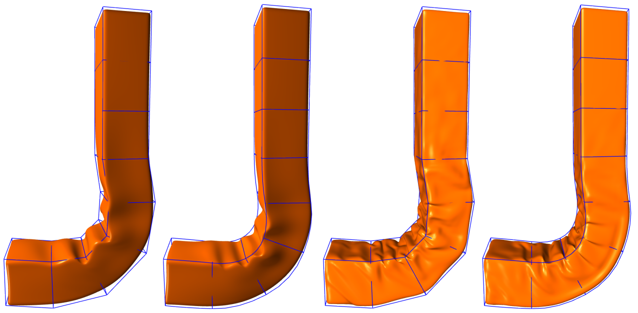

Figure 4: *A comparison of linear and quadratic shape functions for interior elements showing that linear shape functions produce artifacts both at low wrinkle frequencies (left) where wrinkles fall at element boundaries, and at high wrinkle frequencies (right) where the C^0 continuity at interior element boundaries are evident.*

We use non-inverting rotated linear energy for our hexahedral lattice [Irving et al. 2006], and evaluate using 27 quadrature points (3 along each dimension) due to our use of quadratic shape functions. The reason for this is to model smooth global deformations of the interior with a relatively small number of degrees of freedom. Thus, we choose to embed the surface mesh in a coarse regular hexahedral lattice of second order B-Spline elements. Because the coupling between the interior and the surface relies on a frequency filtering, the wrinkles will be influenced by non-smooth deformations of our interior discretization. With linear shape functions, wrinkles driven by a coarse discretization will form at element boundaries due to pinching or discontinuous surface tangents. This artifact is absent with quadratic shape functions because they are C^1 continuous. Figure 4 shows a demonstration of the artifacts that are produced when using linear shape functions in the interior.

5.1 Interior Construction

When building the coarse model, we use a spatial hash to determine the element to which a vertex belongs; however, care must be taken not to lump close yet mechanically separate parts of the mesh into the same element. This is critical, for example, to deal with lips in face geometry. We use superimposed elements with node duplication to accommodate these situations, and briefly describe below how we construct the model given our higher order shape functions.

We automatically obtain a volume model from a watertight mesh. First the mesh faces are divided into lattice cells. Then for each lattice cell, the interior volume mesh is constructed by closing the divided mesh pieces along the cell boundaries. Face normals determine the side of the closed mesh segments to which the interior volume lies. We create a finite element for every disconnected component in each cell. Connectivity of the lattice elements is available by analyzing the connectivity of the closed mesh segments. All elements start with their own copies of their 27 degrees of freedom, 3 quadratic B-spline nodes along each dimension (this would be 8 for trilinear shape functions with hexahedral elements). When two elements are adjacent across a cell face and are topologically connected, we merge the 18 overlapping pairs of quadratic B-spline nodes. Merging nodes ensures C^1 continuity across the common boundary between these elements (or C^0 continuity when merging

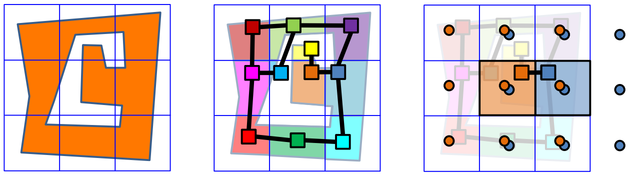

Figure 5: *A 2D illustration of lattice construction. Left, the mesh is sliced along the lattice boundaries. Middle, the mesh segments are closed using the lattice cell faces, a finite element denoted by a colored square is created for each, and connectivity across cell faces forms a graph. Right, visiting each edge of the graph, we merge overlapping nodes in order to mechanically connect the elements (6 pairs of quadratic nodes in 2D, and this would be 18 in 3D).*

4 pairs with trilinear shape functions). Figure 5 shows a 2D illustration of the construction process. This bottom up approach of walking each edge of the topology graph and merging provides a very simple method for assigning the appropriate degrees of freedom to the elements, even in very complicated meshes.

6 Simulation

Wrinkles tend not to have visible dynamics because the skin is thin and light in comparison to its stiffness. Being light and stiff, solving the shell dynamics requires small steps or implicit methods for stability. But the shell stiffness projected onto the interior degrees of freedom is much lower, allowing a wider variety of integration methods and step sizes. Thus we choose to solve the coupled system in a two-step procedure, first computing a quasi-static equilibrium for the shell from the interior configuration q_k at time step k, and then solving the dynamics of the interior while taking into account the shell deformation forces (for instance, with symplectic Euler, or with backward Euler as we describe below). Although skin motion is typically dominated by interior dynamics, our approach cannot produce the high resolution surface dynamics that would appear in a fully dynamic simulation, such as traveling surface waves.

We denote the surface and interior forces as

$$F_x(x) = f_x - \left.\frac{\partial E_x}{\partial x}^T\right|_x , \tag{7}$$

$$F_q(q) = f_q - \left.\frac{\partial E_q}{\partial q}^T\right|_q . \tag{8}$$

Here, $F_x(x)$ is the force acting on the shell at configuration x, and is a combination of external forces f_x and internal forces due to its deformation energy $E_x(x)$. Likewise, $F_q(q)$ is the force acting on the interior at configuration q, and combines external forces f_q such as gravity, and internal forces due to its deformation energy $E_q(q)$. Note that we simulate contact at the level of the embedded thin shell and resolve collision with the iterative robust collision technique of Bridson et al. [2002].

6.1 Surface Statics

The static solution of the shell is constrained by the position of the embedded mesh $x_{qk} = Bq_k$, given the interior configuration q_k at time step k. We use Newton-Raphson iteration to solve for x in the non-linear constrained problem $F_x(x) = 0$ subject to $Hx = Hx_{qk}$, using x_{k-1} as a starting location. We enforce the constraint using Lagrange multipliers,

$$\underset{x,\lambda}{\text{solve}} \left\{ \begin{array}{rcl} F_x + H^T\lambda & = & 0 \\ \gamma H(x_{qk} - x) & = & 0 \end{array} \right. , \tag{9}$$

and we introduce the scalar γ as a simple and inexpensive form of preconditioning to facilitate the solver progression. The value of γ is chosen to equal the shell thickness times its Young's modulus in order to balance the residuals of both equations. This scaling factor is necessary because H contains small values, whereas the stiffness of the shell typically puts F_x at several orders of magnitude larger. At every Newton-Raphson step, the equations are linearized and organized into

$$\left[\begin{array}{cc} \left.\frac{\partial^2 E_x}{\partial x^2}\right|_x & \gamma H^T \\ \gamma H & 0 \end{array} \right] \left[\begin{array}{c} \Delta x \\ \lambda \end{array} \right] = \left[\begin{array}{c} -F_x(x) \\ \gamma H(x_{qk} - x) \end{array} \right] . \tag{10}$$

We find that the solution of this system can be reliably computed with MINRES [Paige and Saunders 1975], where the method of conjugate gradients fails due the non positive definite nature of the system matrix.

We augment our constrained Newton-Raphson solver with a line search at each step using the Armijo rule. We search for step size β_m that satisfies the criteria $\|F(x+\beta_m d)\| < (1-\alpha\beta_m)\|F(x)\|$, for search direction d from current solution x. We let $\alpha = 10^{-4}$, and note that F combines both our non-linear F_x and the constraint $\gamma H(x_{qk}-x)$. We first try β_m equal to 1 and 0.5, and then combined with the value at 0 we build a quadratic polynomial $\phi(\beta) = \|F(x+\beta d)\|^2$ that models the squared error. Every subsequent step finds the minimum β^* of the function ϕ, clamps the guess to 0.1 to 0.5 times the size of the previous best guess, and then tests with the criteria. The process repeats while β^* does not satisfy the criteria, with the model ϕ reconstructed at each step using the last two β and the value at zero (see [Kelley 1987]).

By searching for a static solution that is close to our previous configuration we promote temporal coherence in our animations. While we do not find the closest root to x_{k-1}, this iterative approach is fast and works well for many different step sizes.

6.2 Interior Dynamics

The static solution x_k has resultant force $F_x(x_k) \in \mathbb{R}^n$ for the given interior configuration q_k. When we step the dynamics, we inject the effect of the shell into the interior model by applying the shell forces, using B^T to map shell forces back to the degrees of freedom of the interior. To advance the state of the interior by time step h, we use an implicit scheme, combining both the interior and skin forces,

$$M\frac{\dot{q}_{k+1} - \dot{q}_k}{h} - F_q(q_{k+1}) - B^T F_x(x_{k+1}) \quad = \quad 0 , \tag{11}$$

$$\frac{q_{k+1} - q_k}{h} - \dot{q}_{k+1} \quad = \quad 0 . \tag{12}$$

We must linearize both force terms to perform Newton-Raphson iteration. Interior forces at q_{k+1} are easily approximated by

$$F_q(q_{k+1}) \quad \approx \quad F_q(q_k) - \left.\frac{\partial^2 E_q}{\partial q^2}\right|_{qk} (q_{k+1} - q_k) , \tag{13}$$

but the shell forces are more complicated because x_{k+1} is a function of q_{k+1}. We can make this tractable by using the embedded mesh motion as an approximation of the shell motion,

$$\begin{aligned} F_x(x_{k+1}) \quad &\approx \quad F_x(x_k) + \left.\frac{\partial F_x}{\partial x}\right|_{xk} (x_{k+1} - x_k) , \\ &\approx \quad F_x(x_k) - \left.\frac{\partial^2 E_x}{\partial x^2}\right|_{xk} B(q_{k+1} - q_k) . \end{aligned} \tag{14}$$

These equations are rearranged into a linear system that we solve with Newton-Raphson iterations, using the previous velocities $\dot{q}_k$ as initial guess for $\dot{q}_{k+1}$. At every linearization, we solve the system

$$\begin{aligned} &\left[\ M + h^2 K_q(q_k{+}h\dot{q}_{k+1}) + h^2 B^T K_x(x_k) B\ \right] \left[\ \Delta\dot{q}\ \right] \\ &= h\left(F_q(q_k{+}h\dot{q}_{k+1}) + B^T\left(F_x(x_k) - hK_x(x_k)B\dot{q}_{k+1}\right)\right) \end{aligned} \quad (15)$$

where K_q is the Hessian of the interior energy and K_x is the Hessian of the shell energy. The system is sparse and symmetric and can be solved with conjugate gradients. After each iteration we update the interior velocities $\dot{q}_{k+1} \leftarrow \dot{q}_{k+1} + \Delta\dot{q}$. Notice that we recompute a new Hessian K_q at $(q_k + h\dot{q}_{k+1})$ at each iteration, while the shell stiffness K_x remains fixed.

Once the solver converges to the final velocity, we can set the interior positions $q_{k+1} = q_k + h\dot{q}_{k+1}$, and compute the static shell pose from the embedded mesh position $x_{qk+1} = Bq_{k+1}$ as described in the previous subsection.

We note that our step sizes must not be too large if we wish the static shell solutions to be coherent along an animation sequence. Nevertheless, seeding our static solves at the previous state has worked well with the objects and materials we have simulated using a step size of 0.02 s.

While damping is trivial to add to our formulation above, we can often get sufficient numerical damping from our implicit integration. Our focus is on wrinkling by coupling a quasi-static shell with a dynamic interior, and we note that alternative numerical integrators are preferable should higher fidelity dynamics be desired.

7 Discussion

In the following subsections, we present a variety of examples and comparisons that help validate our approach. This includes informal evaluation of natural phenomena, and comparison of wrinkling wavelengths generated by our model with different parameters to the expected behavior. Table 1 shows information and timing for a variety of models and simulations that appear in figures throughout the paper.

7.1 Full Simulation Comparison

As part of our validation, we compare the results of our embedded thin shell technique to a volumetric simulation of a slab discretized with a large number of volumetric elements. An example simulation of our model and the dense volumetric simulation can be seen at right in Figure 6, while at left we compare wrinkle wavelengths for a large variety of different skin thicknesses and Young's modulus ratios. The different colours of the dots indicate different thicknesses, and we note that the models match well. Note that our wrinkle wavelength tests are limited by the size of the block and the number of the elements. Wrinkles will not form when the wavelength approaches the width of the slab, and at high frequencies the wrinkles take on a wavelength dictated by the lattice resolution.

7.2 Critical Wavelength Comparison

Research on wrinkling often revolves around the modeling of the critical wavelength, such as the wavelength model of Equation 3 given thickness and elasticity parameters. We show in Figure 7 that our embedded thin shell model matches the prediction very well. The figure shows isocontours of equal wavelength for changes in surface thickness and Young's modulus ratio. Measured wavelength samples are shown with grey dots, and involve running a simulation similar to that shown in top right of Figure 6.

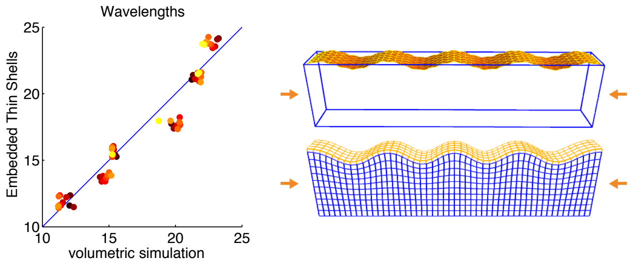

Figure 6: *Left, wrinkle wavelengths compared across many material parameters. Wavelengths are measured from simulations of a thin shell embedded into a single element (top right), and from simulations of dense volumetric elements (bottom right).*

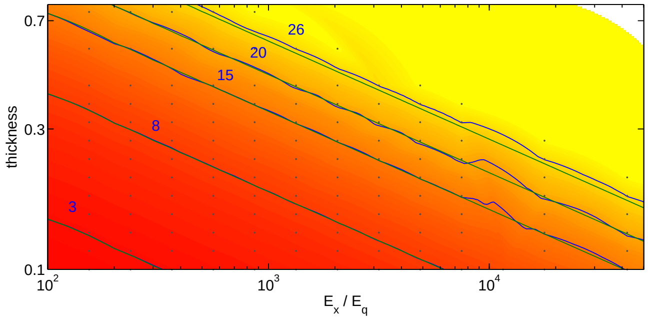

Figure 7: *Wavelengths for different material parameters, as estimated from sample simulations (at grey dots). Wavelength isocontours (blue) match predictions of critical wavelength (green).*

7.3 Natural Phenomena

Although wrinkling in thin sheets has been studied for decades, the understanding of the phenomenon is still far from complete. Multidirectional forces acting on an object and resulting in complex wrinkle morphologies are still rather difficult to fully predict. For example, wrinkling in dried fruits occurs typically at random in-plane because the foundation shrinks isotropically. However, by considering biaxial compression of a thin rigid skin resting on top of an elastic foundation, Mahadevan and Rica [2005] demonstrate that patterns such as Miura-ori form in a two-step process, where buckles form initially without any in-plane orientation. Deformation of these buckles due to a second compression along the buckling direction produces zigzags, also known as a herringbone pattern. Our technique reproduces this phenomenon when a model is subjected to this same compression sequence. An example is shown at top left in Figure 8 with a real example below. The figure also shows that our model produces the familiar flower patterns that are often seen in upholstered furniture.

7.4 Facial Wrinkles

One possible application for embedded thin shells is to produce wrinkles for character animation. Figure 9 shows wrinkles produced for different skin properties, resulting in different wavelengths. In this example, we compute muscle activations for a lattice in which the face is embedded, and estimate these muscle activations from a blend shape deformation model. This allows the interior to undergo novel and dynamic deformations, as opposed to the results which could be produced by using static blend shapes to drive the shell in lieu of embedded mesh positions.

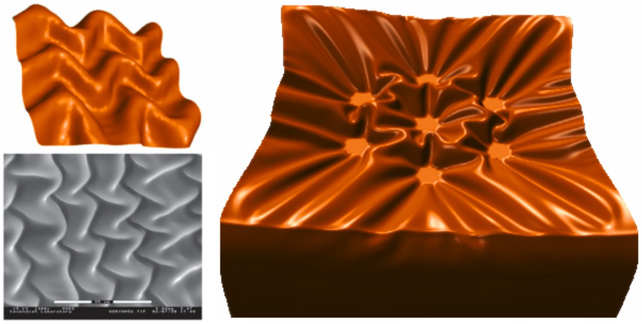

Figure 8: *Left top, a Miura-ori pattern emerging after biaxial compression, and bottom, the same pattern in a thin film atop a thick elastic substrate that manifest in a drying slab of gelatin with a thin skin that forms naturally (from Mahadevan and Rica [2005] with permission). Right, distinctive flower patterns emerge with localized pressure, resembling deep buttoning on upholstered furniture.*

8 Conclusion

Our approach of embedding thin shells can be used to simulate a variety of different kinds of objects with soft interior and a harder skin. Our approach is straightforward and uses intuitive frequency limited constraints to tie the high resolution shell to a low resolution dynamic simulation of the interior deformations. The key is to identify the critical wavelength of wrinkles based on the model parameters, and use this in the design of the constraints. The result is fine details of wrinkles on the surface with a simplified cost of simulating the large geometric deformations of the interior. Furthermore, the quadratic shape functions used for the interior let us keep the number of interior elements needed to a minimum, while avoiding all the C^0 artifacts that linear shape functions produce at element boundaries. In future work, we plan to explore improvements in our static solver, such as preconditioning, and line search modifications. Finally, our largest examples will also benefit from partitioning and parallelization techniques.

Acknowledgments

This work was supported in part by NSERC, the GRAND NCE, and software donation from Autodesk. We thank Sheldon Andrews for collision detection code, and Eitan Grinspun for early discussions at the Bellairs Research Institute.

References

Baraff, D., and Witkin, A. 1998. Large steps in cloth simulation. In *SIGGRAPH '98*, 43–54.

Bergou, M., Wardetzky, M., Harmon, D., Zorin, D., and Grinspun, E. 2006. A quadratic bending model for inextensible surfaces. In *Symposium on Geometry Processing*, 227–230.

Bergou, M., Mathur, S., Wardetzky, M., and Grinspun, E. 2007. Tracks: toward directable thin shells. *ACM Trans. Graph. 26*, 3 (July), 50:1–50:10.

Blinn, J. F. 1978. Simulation of wrinkled surfaces. *SIGGRAPH Comput. Graph. 12*, 3 (Aug.), 286–292.

Bridson, R., Fedkiw, R., and Anderson, J. 2002. Robust treatment of collisions, contact and friction for cloth animation. In *SIGGRAPH '02*, 594–603.

Bridson, R., Marino, S., and Fedkiw, R. 2003. Simulation of clothing with folds and wrinkles. In *Symposium on Computer Animation*, SCA '03, 28–36.

Capell, S., Green, S., Curless, B., Duchamp, T., and Popović, Z. 2002. Interactive skeleton-driven dynamic deformations. *ACM Trans. Graph. 21*, 3, 586–593.

Danielson, D. 1973. Human skin as an elastic membrane. *Journal of Biomechanics 6*, 5, 539–546.

Faloutsos, P., van de Panne, M., and Terzopoulos, D. 1997. Dynamic free-form deformations for animation synthesis. *IEEE Trans. on Vis. and Comp. Graph. 3*, 3, 201 –214.

Genzer, J., and Groenewold, J. 2006. Soft matter with hard skin: From skin wrinkles to templating and material characterization. *Soft Matter 2*, 4, 310–323.

Grinspun, E., Hirani, A. N., Desbrun, M., and Schröder, P. 2003. Discrete shells. In *Symp. on Comp. Anim.*, 62–67.

Irving, G., Teran, J., and Fedkiw, R. 2004. Invertible finite elements for robust simulation of large deformation. In *Symp. on Comp. Anim.*, Eurographics Association, 131–140.

Irving, G., Teran, J., and Fedkiw, R. 2006. Tetrahedral and hexahedral invertible finite elements. *Graph. Mod. 68*, 2, 66–89.

Jimenez, J., Echevarria, J. I., Oat, C., and Gutierrez, D. 2011. *GPU Pro 2*. AK Peters Ltd., ch. Practical and Realistic Facial Wrinkles Animation, 15–27.

Kavan, L., Gerszewski, D., Bargteil, A. W., and Sloan, P.-P. 2011. Physics-inspired upsampling for cloth simulation in games. *ACM Trans. Graph. 30*, 4 (July), 93:1–93:10.

Kelley, C. 1987. *Solving Nonlinear Equations with Newton's Method*. Fundamentals of Algorithms. Society for Industrial and Applied Mathematics.

Larboulette, C., and Cani, M.-P. 2004. Real-time dynamic wrinkles. In *Computer Graphics International*, 522–525.

Lévy, B., and Liu, Y. 2010. Lp centroidal Voronoi tessellation and its applications. *ACM Trans. Graph. 29*, 4, 119:1–119:11.

Magnenat-Thalmann, N., Kalra, P., Luc Leveque, J., Bazin, R., Batisse, D., and Querleux, B. 2002. A computational skin model: fold and wrinkle formation. *IEEE Trans. Information Technology in Biomedicine 6*, 4, 317–323.

Mahadevan, L., and Rica, S. 2005. Self-organized origami. *Science 307*, 5716, 1740–1740.

McAdams, A., Zhu, Y., Selle, A., Empey, M., Tamstorf, R., Teran, J., and Sifakis, E. 2011. Efficient elasticity for character skinning with contact and collisions. *ACM Trans. Graph. 30*, 4, 37:1–37:12.

Mezger, J., Thomaszewski, B., Pabst, S., and Strasser, W. 2009. Interactive physically-based shape editing. *Computer Aided Geometric Design 26*, 6, 680–694.

Molino, N., Bao, Z., and Fedkiw, R. 2004. A virtual node algorithm for changing mesh topology during simulation. *ACM Trans. Graph. 23*, 3, 385–392.

Müller, M., and Gross, M. 2004. Interactive virtual materials. In *Proceedings of Graphics Interface 2004*, GI '04, 239–246.

Müller, M., Dorsey, J., McMillan, L., Jagnow, R., and Cutler, B. 2002. Stable real-time deformations. In *Symposium on Computer Animation*, SCA '02, 49–54.

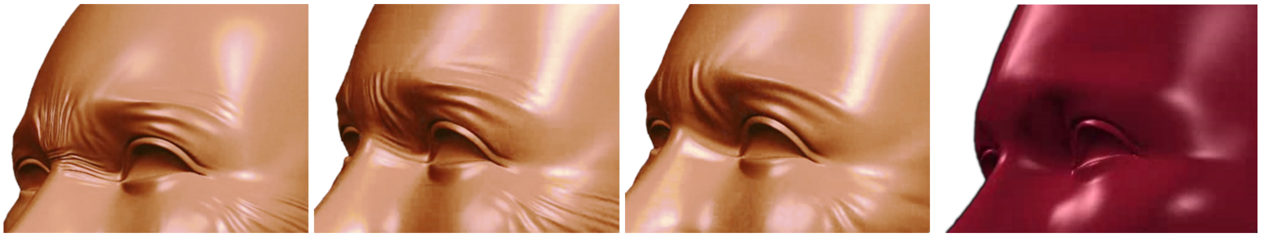

Figure 9: *Wrinkles on a face produced by muscles embedded in the FEM volume and activated to create the motion. Results are shown with 3 sets of skin stiffness and bending paramenters (left) along with the large deformation embedded mesh (right).*

	h	$\mathsf{E}_x/\mathsf{E}_q$	λ	c	n	n_x	n_q	$t_{q\text{-setup}}$	$t_{H\text{-setup}}$	t_x	t_{xA}	t_{Fx}	t_q	t_{qA}	t_{Fq}
pillow	0.0002	10^2	0.00404	2108	28194	19792	45	1584	1377	45690	521	14	45	5	4
pillow	0.0004	10^2	0.00808	1488	12534	8352	45	982	1231	6328	303	6	37	4	3
pillow	0.0005	10^2	0.01010	1046	3138	2088	45	361	329	639	47	1	35	4	2
pillow	0.0007	10^4	0.06566	98	28194	19792	45	1584	7902	36746	502	14	45	5	4
pillow	0.0007	10^4	0.06566	100	12534	8352	45	982	1128	4536	219	5	37	4	3
pillow	0.0007	10^4	0.06566	95	3138	2088	45	361	234	576	40	1	35	4	2
pillow	0.0007	10^4	0.06566	87	636	420	45	163	89	69	7	1	34	3	2
bar	0.001	10^2	0.02021	33	6534	4352	8	280	805	2257	189	6	6	0.4	0.3
bar	0.0001	10^2	0.00202	82	6534	4352	8	296	649	2694	196	5	6	0.4	0.3
slab	0.001	10^2	0.02021	6	918	500	1		109	162	12	0.7	1	0.1	0.1
slab	0.0002	10^3	0.00870	21	918	500	1		108	252	19	0.7	1	0.1	0.1

Table 1: *Example sizes and times per frame for some of different simulations involving the 50 cm pillow in the supplementary video, the 80 cm bar from Figure 3, and the 50 cm slab from Figure 6. The skin thickness is h meters, the Young's Modulus of skin and interior are E_x and E_q respectively, λ is the critical wavelength in meters, c is the number of constraints, n the number of shell degrees of freedom, n_x is the number of faces in the mesh, n_q is the number of hexahedral elements, $t_{q\text{-setup}}$ is the lattice setup time, $t_{H\text{-setup}}$ is the constraint setup time, t_x is the time to linearize, assemble, solve and update the system, t_{xA} is the assembly portion of the static solve time, t_{Fx} is the time to compute the shell forces. Finally t_q is the time to linearize assemble solve and update the dynamic system, t_{qA} is assembly portion of the dynamic solve time, and t_{Fq} is the time to compute lattice elastic forces. All times are in milliseconds.*

Nesme, M., Kry, P. G., Jeřábková, L., and Faure, F. 2009. Preserving topology and elasticity for embedded deformable models. *ACM Trans. Graph. 28*, 3, 52:1–52:9.

Paige, C., and Saunders, M. 1975. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis 12*, 4, 617–629.

Parker, E. G., and O'Brien, J. F. 2009. Real-time deformation and fracture in a game environment. In *Symposium on Computer Animation*, SCA '09, 165–175.

Rohmer, D., Popa, T., Cani, M.-P., Hahmann, S., and Sheffer, A. 2010. Animation wrinkling: augmenting coarse cloth simulations with realistic-looking wrinkles. *ACM Trans. Graph. (SIGGRAPH ASIA) 29*, 6 (Dec.), 157:1–157:8.

Sederberg, T. W., and Parry, S. R. 1986. Free-form deformation of solid geometric models. *SIGGRAPH Comput. Graph. 20*, 4 (Aug.), 151–160.

Seiler, M., Spillmann, J., and Harders, M. 2012. Enriching coarse interactive elastic objects with high-resolution data-driven deformations. In *Symp. on Comptuer Animation*, 9–17.

Sifakis, E., Neverov, I., and Fedkiw, R. 2005. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Trans. Graph. 24*, 3 (July), 417–425.

Sifakis, E., Shinar, T., Irving, G., and Fedkiw, R. 2007. Hybrid simulation of deformable solids. In *Symposium on Computer Animation*, 81–90.

Terzopoulos, D., and Waters, K. 1990. Physically-based facial modelling, analysis, and animation. *The journal of visualization and computer animation 1*, 2, 73–80.

Terzopoulos, D., and Witkin, A. 1988. Physically based models with rigid and deformable components. *IEEE Comput. Graph. Appl. 8*, 6 (Nov.), 41–51.

Terzopoulos, D., Platt, J., Barr, A., and Fleischer, K. 1987. Elastically deformable models. *SIGGRAPH Comput. Graph. 21*, 4 (Aug.), 205–214.

Thomaszewski, B., Wacker, M., and Strasser, W. 2006. A consistent bending model for cloth simulation with corotational subdivision finite elements. In *Symposium on Computer Animation*, SCA '06, 107–116.

Timoshenko, S., and Gere, J. 2009. *Theory of Elastic Stability*. Dover Civil and Mechanical Engineering Series. Dover.

Venkataraman, K., Lodha, S., and Raghavan, R. 2005. A kinematic-variational model for animating skin with wrinkles. *Computers & Graphics 29*, 5, 756 – 770.

Wang, Y., Wang, C. C., and Yuen, M. M. 2006. Fast energy-based surface wrinkle modeling. *Comp. Graph. 30*, 1, 111 – 125.

Wojtan, C., and Turk, G. 2008. Fast viscoelastic behavior with thin features. *ACM Trans. Graph. 27*, 3 (Aug.), 47:1–47:8.