Non-Linear Mechanics and Collisions for Subdivision Surfaces

Eitan Grinspun Fehmi Cirak Peter Schröder Michael Ortiz

Caltech*

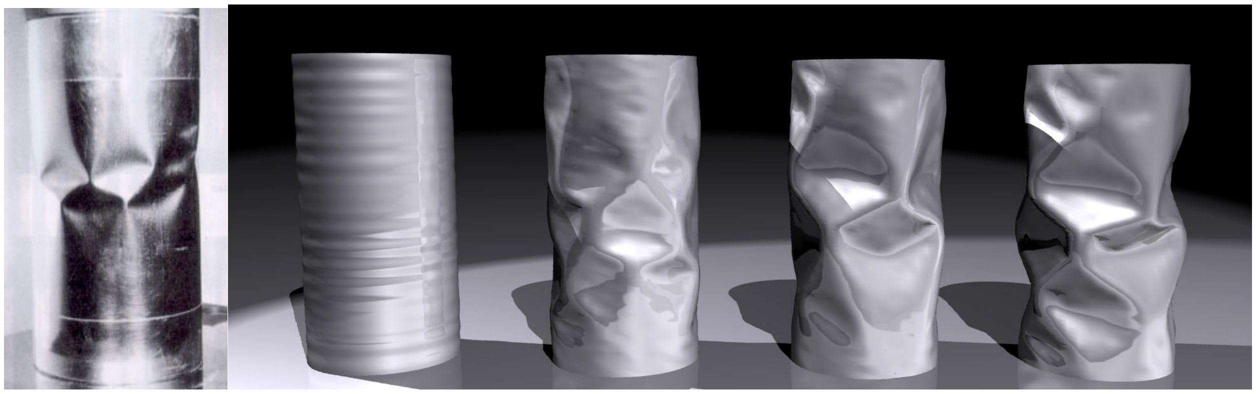

Figure 1: *Comparison of empirical and simulated results.*

Abstract

Numerically accurate simulation of the mechanical behavior of thin flexible structures is important in application areas ranging from engineering design to animation special effects. Subdivision surfaces provide a unique opportunity to integrate geometric modeling with concurrent finite element analysis of thin flexible structures. Their mechanics are governed by the so-called thin-shell equations. We present a concise treatment of thin-shell equations including dynamic behavior, scalable material models, and the treatment of collisions (detection as well as response). The resulting energy minimization problem is non-linear and in turn able to capture effects of far more realism than linear models. We demonstrate these claims with a number of simulations which exhibit characteristic effects of real world experiments.

1 Introduction

Subdivision is an attractive method for free form geometric modeling. It elegantly addresses the classic challenge of building (piecewise) smooth surfaces of arbitrary topology [7, 10] and dovetails nicely with many modern algorithms which exploit multiresolution (for an overview see [32]). In applications such as engineering design and animation it is also necessary to model the (dynamic or static) mechanical response of such surfaces subject to appropriate material models and external forces.

The mechanical behavior of thin flexible surfaces is described most naturally by the so called thin-*shell* equations based on the classic Kirchhoff-Love theory [28]. It describes the response of the surface to external forces in terms of intrinsic quantities such as the first and second fundamental forms of the original and deformed surface. As such it leads to a non-linear energy minimization problem involving as highest order quantities curvatures. The straight forward finite element approach to solving these equations requires basis functions with square integrable curvatures. Owing to their smoothness properties, subdivision schemes provide an ideal basis for an accurate and consistent finite element treatment of thin-shell equations [2, 1]. Subdivision schemes *simultaneously* provide very flexible representations for the geometry, particularly in the arbitrary topology surface setting.

Thin-*shell* equations are closely related to thin-*plate* equations, which are commonly used in computer graphics and animation. The thin-plate setting assumes that the original geometry is flat. Consequently, first and second order derivatives decouple and the associated energy minimization functional is reduced to a simple linear combination of weighted first and second *derivative* squared norms of the surface. While this formulation is very useful for variational geometric modeling [8, 11, 31] and intuitive direct manipulation of surfaces [24, 23, 27], it cannot capture subtleties of the nonlinear dynamic behavior of more complex shapes. These nonlinearities are particularly important for the accurate modeling of stability phenomena, which occur, among other situations in the simulation of crushing. For example, the crushing of a sheet of paper or a car body.

This difference between linear and non-linear treatments is exemplified in Figure 2. A simple metal cylinder is clamped at the ends and two opposite concentrated forces are pulling it apart. The left image shows the result of the full non-linear thin-shell treat-

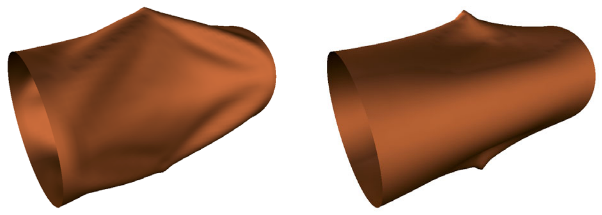

Figure 2: *A metal cylinder is clamped at the ends and pulled apart by concentrated forces. On the left the full non-linear thin-shell treatment. On the right a linearized version of the thin-shell model.*

*{eitan,cirak,ps}@cs.caltech.edu, ortiz@atlantis.caltech.edu

ment, while the right image shows the result of a linear treatment.

Goals and Contributions Robust and mechanically accurate simulation of thin flexible structures based on classic Kirchhoff-Love theories has been a long standing and difficult problem in the mechanical engineering research community. Particularly in the arbitrary topology setting many numerical techniques fail catastrophically (e.g., "shear locking"). Subdivision surfaces and the basis ("shape") functions induced by subdivision have recently been demonstrated to offer superior performance and robustness in the finite element treatment of thin-shell equations [2, 3], entirely avoiding the shear locking problem. We build directly on the research reported in [2, 3] and extend in a number of new directions. Novel elements of the research presented in the present paper are

- the treatment of subdivision surface collisions covering detection and response;
- the use of finite element shape functions based on a more sophisticated variant of Loop's [20] scheme, which accurately models boundaries, creases, and convex as well as concave corners [5];
- a novel quadrature method for the limit surface integrals appearing in the finite element treatment, which elegantly deals with all possible combinations of feature tags (boundaries/creases/corners);
- a scalable approach to mechanics simulation with a common framework for applications ranging from qualitative animation to quantitative engineering design.

Even though the resulting numerical computations are non-linear and costly the increase in mechanical accuracy and realism is well worth the extra computational effort. Figure 8 shows several time steps from a simulation we performed with our algorithm.

In the remainder of this section we briefly review related work. Section 2 presents a concise and flexible treatment of the kinematic, constitutive, and dynamic equations for thin flexible structures, including treatment of collisions. The focus is on formulations which accomodate various material and collision response models. More sophisticated (and perhaps costly) models are easily interchanged with simplified models. Sections 3 and 4 describe efficient (self-)collision detection algorithms and a novel quadrature method for the subdivision limit functionals required by the finite element method. Finally, we demonstrate our complete simulation framework with a number of examples.

1.1 Review of Related Work

In this section we briefly review related work and delineate our contributions vis-a-vis existing work more precisely. It falls into three main categories: (a) thin-shells, (b) subdivision surfaces, and (c) collision detection, which we treat in turn.

Simulation of Thin-Shells is notoriously sensitive to ill-conditioning due to the different orders of the dimensions in the thickness and length directions (see e.g., [15]). Methods which are numerically simple require smooth shape functions, i.e., those which posses square integrable curvatures. Conventional finite element approaches, i.e., those based on piecewise linear finite elements try to avoid the smoothness requirement with cumbersome modifications. However, utilizing subdivision surfaces the smoothness requirement is easily fulfilled. In [2, 3] we introduced a novel method based on subdivision surfaces which elegantly sidesteps all the difficulties inherent to the conventional approaches. The treatment in [2] focused on demonstrating the basic method through passing the so-called "obstacle course" of benchmark problems, while [3] was devoted to large deformation analysis. This work is extended here principally through the inclusion of collision detection and response, and a novel quadrature method.

Subdivision Surfaces have been used as finite elements in the context of variational modeling. Halstead et al. [12] used a linear thin-plate functional for Catmull-Clark surface fairing under interpolation constraints, while [22, 21] employed linear thin-plate functionals for an intuitive modeling interface and shape reconstruction. Since such functionals depend on the parameterization used, difficulties such as infinite bending energies occur in the vicinity of extraordinary vertices. Analytical and numerical convergence studies show that these problems do not appear for thin-*shells*, as their associated energy functionals involve integrals of *intrinsic* surface properties only [1, 2]. An important element in such numerical treatments is the reliable and accurate evaluation of limit surface properties to compute the finite element integrals. We do not require the full power of arbitrary parameter evaluation [26], but instead only require the values at selected parameter locations. We give a simple and flexible quadrature method which expresses all required quantities as simple linear combinations of coarsest level control points. No actual subdivision code (with quadtree structures, etc.) is required. In particular our method can be used to "retrofit" any existing triangle based FEM code and allow it to use Loop shape functions.

Collision Detection and Response are an important component of dynamics simulation. The problem of interference detection has been studied extensively (for an excellent survey see [19]). In general, the most effective approaches employ a hierarchical bounding volume structure to avoid expensive intersection tests between complex objects. Of particular interest to us is self-collision detection for subdivision surfaces. A self-collision detection algorithm differs from more general algorithms in that it is not confused by the intersection at the seam of two adjacent patches. Hughes et al. [14] describe an accurate algorithm for objects undergoing polynomial deformation. The algorithm uses linear programming, subdivision trees, sweep-and-prune [9], and loop detection [13] to check for interference. The implementation cost is significant. Volino and Thalmann present an algorithm intended for polygonal surfaces undergoing dynamic deformation [30, 29]. They create a patch hierarchy, use surface-curvature tests to rule out self-intersections, and use a combination of surface-curvature and bounding box tests to rule out intersection between pairs of subsurfaces. We extend this approach to the domain of subdivision surfaces. This involves finding a bound on the gauss map of a subdivision patch, and finding a tight, efficient bounding volume for the patch. We also extend the approach by using sweep-and-prune and tighter bounding volumes for the patches in the hierarchy.

When interference is detected, several techniques based on gap functions exist in order to constrain or penalize the interpenetration. Commonly used methods have a number of shortcomings, most prominently assymetrical treatment of the two interfering surfaces. Additionally they tend to introduce numerical stiffness [4]. Our approach is rooted in recently proposed variational methods for nonsmooth contact [16] and avoids these difficulties.

2 Mechanics of Thin-Shells

In this section we summarize the field equations for thin-shells. Our presentation is restricted to the Kirchhoff-Love model. The reader interested in thin-shell models is referred to [2] for pointers to the specialized literature.

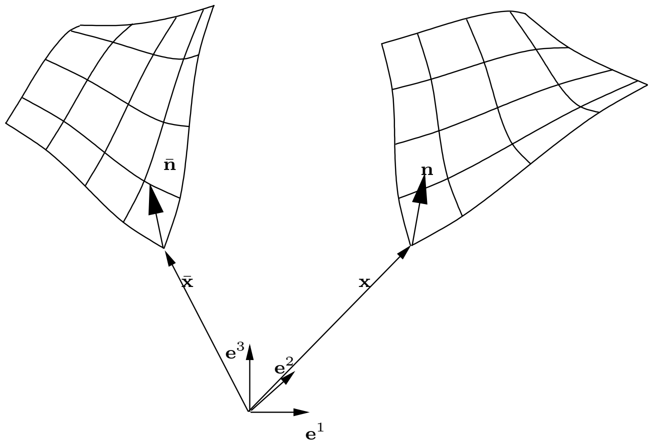

Figure 3: *Shell middle surface in the reference and the deformed configurations.*

2.1 Kinematics and FEM Discretization of the Stationary Case

The shell is described in its initial and deformed configurations, $\bar{\mathbf{f}}$ and $\mathbf{f}$ (resp.), with reference to a domain $\bar{\Omega}$ and boundary $\bar{\Gamma}$ which are induced by the control polyhedron

$$\begin{aligned}\bar{\mathbf{f}}(u,v,w) &= \bar{\mathbf{x}}(u,v) + w\bar{\mathbf{n}}(u,v)\\ \mathbf{f}(u,v,w) &= \mathbf{x}(u,v) + w\lambda(u,v)\mathbf{n}(u,v),\end{aligned}$$

where $(u,v) \in \bar{\Omega}$ are the parameters of the middle surface $\bar{\mathbf{x}}$ (resp. $\mathbf{x}$), $w \in [-\frac{1}{2}\bar{h}, +\frac{1}{2}\bar{h}]$ ranges over the small thickness $\bar{h}(u,v)$, $\lambda = h/\bar{h} > 0$ describes the thickness change, $\bar{\mathbf{n}}$ (resp. $\mathbf{n}$) denote the surface normals, which follow as the normalized cross product of the first two partial derivatives of the middle surface. To simplify the following derivations we will assume $\lambda = 1$ (see [3] for a derivation including the thickness parameter as an explicit degree of freedom for the simulation).

The middle surface is given as a linear combination of subdivision basis functions with control points as coefficients $\bar{\mathbf{x}} = \sum_{i=0}^{m} \bar{\mathbf{p}}_i B^i(u,v)$, where i ranges over all control points and B^i denotes the associated basis function. A similar expression holds for the deformed configuration.

The covariant basis vectors are given by the partial derivatives of the surface with respect to the parameters (u,v,w)

$$\bar{\mathbf{g}}_1 = \bar{\mathbf{x}}_u + w\bar{\mathbf{n}}_u,\ \bar{\mathbf{g}}_2 = \bar{\mathbf{x}}_v + w\bar{\mathbf{n}}_v,\ \bar{\mathbf{g}}_3 = \bar{\mathbf{n}},$$

and similarly for the deformed configuration. Contravariant base vectors follow as usual from the basis duality relations $\bar{\mathbf{g}}^i \cdot \bar{\mathbf{g}}_j = \delta^i_j$. With this notation the metric tensor in its covariant basis (resp. contravariant basis) follows as $\bar{g}_{ij} = \bar{\mathbf{g}}_i \cdot \bar{\mathbf{g}}_j$ and $\bar{g}^{ij} = \bar{\mathbf{g}}^i \cdot \bar{\mathbf{g}}^j$. For later use we also define, as a function of the deformed configuration, the right Cauchy Green deformation tensor $C_{ij} = \mathbf{g}_i \cdot \mathbf{g}_j$.

barred quantity: refers to the initial configuration. e.g., $\bar{\mathbf{f}}$ and $\mathbf{f}$ are the initial and deformed configurations, resp.	
differentiation: $x_u \equiv \frac{\partial x}{\partial u}$ and $x_v \equiv \frac{\partial x}{\partial v}$.	
indexed expr.: $i, j, k, l \in \{1,2,3\}$ (unless contraindicated).	
trace operator: denoted by "$:$". $A : B \equiv \sum_j (AB^T)_{jj}$.	
interval operator: denoted by "$\square$". e.g., $\square B^i \equiv$ range(B^i).	
delta function: $\delta^i_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise}\end{cases}$	

Table 1: *Mathematical notation.*

The potential energy Π of the shell body can now be written as a function of the deformation tensor

$$\Pi(\mathbf{f}) = \Pi_{\text{stat}}(\mathbf{f}) + \Pi_{\text{ext}}(\mathbf{f}), \quad \Pi_{\text{stat}} = \int_{\bar{\Omega}} \int_{-\bar{h}/2}^{\bar{h}/2} W(\mathbf{C})\, d\bar{V},$$

where W denotes the free energy density, which incorporates the material model (see below) and the measure $d\bar{V}$ accounts for the shell curvature in the thickness integration

$$d\bar{V} = \frac{\det(\bar{\mathbf{g}}_1, \bar{\mathbf{g}}_2, \bar{\mathbf{g}}_3)}{\det(\bar{\mathbf{x}}_u, \bar{\mathbf{x}}_v, \bar{\mathbf{n}})}\, dw\, d\bar{\Omega}.$$

Expressing the potential energy of the thin-shell in this form is only one of many choices (see [3] for an alternative formulation in terms of the deformation gradient).

Applying the standard calculus of variations to solve this energy minimization problem, we set the first variation to zero:

$$\delta\Pi = \mathbf{G}_{\text{stat}} + \mathbf{G}_{\text{ext}} = 0, \tag{1}$$

where $\mathbf{G}_{\text{stat}} = \delta\Pi_{\text{stat}}$, and $\mathbf{G}_{\text{ext}} = \delta\Pi_{\text{ext}}$ describes external forces which we discuss later. The resulting weak form is given by

$$\mathbf{G}_{\text{stat}} = \int_{\bar{\Omega}} \int_{-\bar{h}/2}^{\bar{h}/2} \frac{\partial W}{\partial \mathbf{C}} : \delta\mathbf{C}\, d\bar{V} = \int_{\bar{\Omega}} \int_{-\bar{h}/2}^{\bar{h}/2} \mathbf{S} : \delta\mathbf{C}\, d\bar{V},$$

where $\mathbf{S}$ denotes the so-called second Piola-Kirchhoff stress tensor. Substituting the shell description in terms of subdivision basis functions into the expression for $\delta\mathbf{C} = \delta\mathbf{g}_i \cdot \mathbf{g}_j + \mathbf{g}_i \cdot \delta\mathbf{g}_j$ we find

$$\begin{aligned}\mathbf{G}^i_{\text{stat}} &= \int_{\bar{\Omega}} \int_{-\bar{h}/2}^{\bar{h}/2} \quad \mathbf{g}_1^T \mathbf{S} \tfrac{\partial \mathbf{x}_u}{\partial \mathbf{p}_i} + \mathbf{g}_2^T \mathbf{S} \tfrac{\partial \mathbf{x}_v}{\partial \mathbf{p}_i} + \mathbf{g}_3^T \mathbf{S} \tfrac{\partial \mathbf{n}}{\partial \mathbf{p}_i} + \\ &\qquad w(\mathbf{g}_1^T \mathbf{S} \tfrac{\partial \mathbf{n}_u}{\partial \mathbf{p}_i} + \mathbf{g}_2^T \mathbf{S} \tfrac{\partial \mathbf{n}_v}{\partial \mathbf{p}_i})\, d\bar{V},\end{aligned} \tag{2}$$

where i is an index over all control vertices. For each G^i_{stat} a numerical quadrature is performed. We use Simpson's rule over the thickness of the shell and an edge midpoint quadrature for the shell middle surface. Note that the latter needs to incorporate in $d\bar{V}$ the Jacobian $|\bar{\mathbf{x}}_u \times \bar{\mathbf{x}}_v|$ of the mapping from a standard triangle to the actual domain triangle in the control polyhedron ($\bar{\Omega}$). While Equation 2 is relatively straightforward we strongly recommend the use of a computer algebra system to compute all involved quantities.

The external forces are computed as follows

$$\mathbf{G}_{\text{ext}} = -\int_{\bar{\Omega}} \mathbf{q} \cdot \delta\mathbf{x}\, d\bar{\Omega} - \int_{\bar{\Gamma}} \mathbf{N} \cdot \delta\mathbf{x}\, d\bar{\Gamma}$$

where $\mathbf{q}$ is the distributed load per unit area of $\bar{\Omega}$ and $\mathbf{N}$ is the axial force per unit length of the boundary $\bar{\Gamma}$.

2.2 Constitutive Equations

So far we have said nothing about the form $\mathbf{S} = \partial\mathbf{W}/\partial\mathbf{C}$ takes. There are many choices of material models and we briefly discuss a simple model. For more complex models the reader is referred to [3, 25]. The internal energy for a general linear elastic material can be expressed with

$$W = \frac{1}{2}\mathbf{E} : \mathbf{D} : \mathbf{E} = \frac{1}{2}\sum_{ijkl} E_{ij} D^{ijkl} E_{kl}$$

where $\mathbf{D}$ is the fourth order constitutive tensor and $E_{ij} = \frac{1}{2}(C_{ij} - \bar{g}_{ij})$ is the Green-Lagrange strain tensor. For isotropic materials $\mathbf{D}$ takes the form

$$D^{ijkl} = \frac{E\nu}{1-\nu^2}\bar{g}^{ij}\bar{g}^{kl} + \frac{E}{2(1+\nu)}(\bar{g}^{ik}\bar{g}^{jl} + \bar{g}^{jk}\bar{g}^{il})$$

with the two material parameters Young's modulus E and Poisson's ratio ν. The linear relationship between the stresses and strains $\mathbf{S} = \mathbf{D} : \mathbf{E}$ is only valid for sufficiently small strains. It should be noted that large displacements do not always lead to large strains. Especially thin-shells exhibit in general large displacements and small strains (e.g., crumpling of paper).

2.3 Non-Stationary Case

For dynamical simulation we must include the inertia effects G_{dyn} of the surface in the weak form of the equilibrium (Equation 1). Since we are dealing with thin-shells, we neglect the higher order rotational inertia effects. With the mass density ρ we obtain

$$G_{\mathrm{dyn}} = \int_{\bar{\Omega}} \int_{-\bar{h}/2}^{\bar{h}2} \rho\, \ddot{\bar{\mathbf{x}}} \cdot \delta\bar{\mathbf{x}}\, d\bar{V}.$$

The equivalent expression to the weak form Eq. 1 now requires

$$G_{\mathrm{stat}} + G_{\mathrm{ext}} = G_{\mathrm{dyn}}. \tag{3}$$

For the time integration of the second order semi-discretized equations we use Newmark's algorithm [15]. Because we are dealing with strongly nonlinear simulations we utilize its explicit version.

$$\begin{aligned} \mathbf{x}^{(t+dt)} &= \mathbf{x}^{(t)} + dt\,\dot{\mathbf{x}}^{(t)} + dt^2/2\ddot{\mathbf{x}}^{(t)} \\ \dot{\mathbf{x}}^{(t+dt)} &= \dot{\mathbf{x}}^{(t)} + dt(1-\gamma)\ddot{\mathbf{x}}^{(t)} + dt\gamma\ddot{\mathbf{x}}^{(t+dt)}, \end{aligned} \tag{4}$$

with the numerical damping factor $0.5 \leq \gamma \leq 1.0$. Explicit time integration schemes are well suited for strongly nonlinear problems since implicit methods are very expensive and, more importantly, not unconditionally stable as is the case for linear problems. The accelerations follow from Equation 3 with

$$\mathbf{M}\ddot{\mathbf{x}}^{(n)} = (\mathbf{G}_{\mathrm{stat}}^{(t)} + \mathbf{G}_{\mathrm{ext}}^{(t)}),\ M_{ij} = \int_{\bar{\Omega}} \int_{-\bar{h}/2}^{\bar{h}2} \rho B^i B^j\, d\bar{V},$$

and subsequently the displacements and velocities are updated with Equations 4. Mass matrix $\mathbf{M}$ is computed by numerical quadratures as for the other terms (Equation 2), and we use a mass lumping technique ($\mathbf{M}$ is diagonalized by summing entries for each row) to decouple the system.

In the presence of collisions the displacements $\mathbf{x}^{(t+dt)}$ are not always contained in the solution space and have to be corrected. Our correction strategy will be described in Section 3.3. After resolving all possible collisions we obtain the corrected displacements $\mathbf{x}_{corr}^{(t+dt)}$. The accelerations $\ddot{\mathbf{x}}_{corr}^{(t+dt)}$ associated with the corrected displacements follow with

$$\ddot{\mathbf{x}}_{corr}^{(t+dt)} = \ddot{\mathbf{x}}^{(t+dt)} + \frac{2}{dt^2}(\mathbf{x}^{(n+1)} - \mathbf{x}_{corr}^{(n+1)})$$

and subsequently the velocities are updated with the corrected accelerations as usual.

3 Collision Detection and Response

In order to deal with crushing and crumpling collision detection and response are crucial. In this section we discuss our collision detection and response strategies. Since we expect to deal with surfaces which fold over on themselves self interference detection is particularly critical and we begin with its treatment.

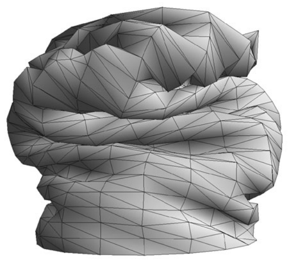
Figure 4: *The collision algorithm must handle difficult buckling situations.*

3.1 Self-Interference Detection

This section describes how we find self-intersections of the subdivision surface. We present a technique applicable to various subdivision schemes, including the Loop scheme.

3.1.1 Detection Algorithm

The limit surface is trivially parameterized over the control mesh triangles, implying a decomposition of the surface into triangular patches called *limit patches*. The goal of the algorithm is to identify all *self*-intersecting limit patches, all *pairs* of intersecting limit patches, and to determine the *intersection curve* up to some specified precision.

The collision detection problem is very difficult. Typically, if there are N objects, there are $O(N^2)$ possible collisions. Furthermore, if we consider a surface composed of patches, there are $O(N)$ pairs of objects that are adjacent and thus apparently colliding, yet the seam between them should not be considered a region of interference. A multiresolution surface representation makes the problem more tractable.

The limit surface is represented as a hierarchy of patches. Every level is a disjoint set of patches that covers the surface, and every patch is a disjoint union of its children (sub-patches). The root consists of one patch, and the leaves are the limit patches. Conceptually, every leaf has an infinite hiearchy of descendants, corresponding to the process of recursive subdivision.

Every patch in the hierarchy has an associated axis-aligned bounding box (AABB), an object-aligned bounding box (OBB), and a bound on its Gauss map (see 3.2). The Gauss map is a function $(u, v) \mapsto (\mathbf{f}_u \times \mathbf{f}_v)/\|\mathbf{f}_u \times \mathbf{f}_v\|_2$ that maps every point on the surface to its oriented unit normal vector [6]. Bounding volumes are used to rule out interference between non-adjacent patches, while the Gauss map bound is used to rule out self interference and interference between adjacent patches.

At the finest level of the hierarchy, a bounding prism is used to provide a tight bound on the geometry of the limit patch. Originally used in the ray-tracing setting by Kobbelt [17], the prism is constructed by sweeping the chord triangle spanned by the three corners of the limit patch in the chord-normal direction.

The procedure for finding all self intersections of the surface consists of three stages:

- A *covering set* of patches is constructed, such that each patch is guaranteed not to self-intersect, and the disjoint union of the patches covers the surface.
- All pairs of intersecting patches in the covering set are discovered.
- The actual intersection curve can be traced using the algorithm in [13] (this is not required for our simulation).

The first two stages are now discussed in detail.

The covering set is constructed by a depth-first traversal of the patch hierarchy. When each patch is visited, a test is made to rule out self-intersection. If the test succeeds, the patch is added to the

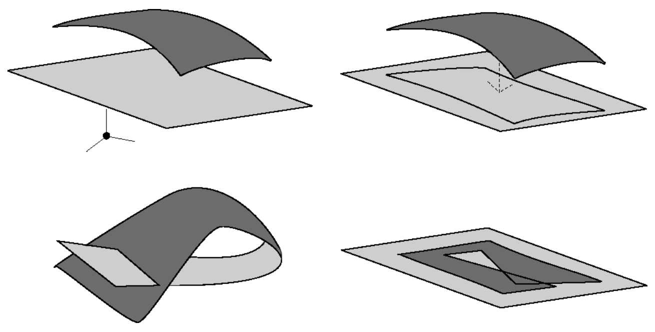

Figure 5: *A sufficient condition that a patch does not self-intersect is the existence of a separating plane between the origin and the Gauss map of the patch (top-left), and the absence of intersections in the projection of the contour of the patch onto the separating plane (top-right). The first condition ensures that the patch does not self-intersect due to high curvature (bottom-left), while the latter ensures that the patch does not self-intersect due to planar stretching and shearing (bottom-right).*

covering set, and its children are pruned from the search. The self-intersection test is similar to the test in [30]. Two conditions must hold for ruling-out self-intersection. First, there must exist a plane separating the Gauss map from the origin (Figure 5 top-left). Second, the projection of the patch boundary onto this plane must lie in the plane and have no self-intersections (Figure 5 top-right). In practice, we have found that the accuracy of the simulation is not affected by omitting this second condition because most collisions occur as a result of bending, not stretching or shearing of the thin-shell.

Instead of rebuilding the covering set at each time-step with a depth-first search, it is possible to use an incremental approach similar to [18]. Once the covering set is updated, the list of colliding patch pairs must be constructed. The sweep-and-prune algorithm is used to incrementally update the list of pairs with overlapping AABBs [9]. If a pair is not on this list it is absolved from further consideration. Remaining pairs are tested for topological adjacency (described below). Interference detection of non-adjacent pairs is performed using their associated OBB hierarchy. The subdivision is terminated when the subpatches are sufficiently planar. At that stage we approximate them with planes and perform an overlap test.

For adjacent pairs in the list, the gauss map test described above is used to rule out intersection (since the union of the pair forms a connected surface). If the test fails, the larger patch is decomposed into its constitutive sub-patches, and these are tested against the smaller patch. Again, prune-and-sweep is used to eliminate pairs based on their AABBs. Those pairs that remain are tested in the same manner described above.

Adjacency Test Every patch has a list of all its immediate neighbors on the same level. To test for adjacency of a pair of patches a and b, where b is at a coarser level of the hierarchy, all the neighbors of a are examined. If b is an ancestor of one of the neighbors, then a and b are adjacent.

Quantization of Gauss Map Bound The Gauss map bound is represented in quantized form by a bit mask. The state of every bit indicates whether the Gauss map is completely contained in a particular halfspace. The half spaces are chosen stochastically from the uniform distribution of half spaces defined by planes passing through the origin. This is easily achieved by sampling the uniform distribution of points on a sphere, and interpreting the result as the (unit-)normal of a plane containing the origin. In practice, 32 samples (32 bits) are more than sufficient.

3.2 Bound on the Gauss Map of a Subdivision Patch

This section describes how to bound the gauss map of a limit patch. The theory is followed by implementation notes and observations.

Theory We assume that the set of extraordinary vertices is independent. Consider a limit patch $\mathbf{x}(u,v)$ and the supporting control vertices $\mathbf{p}_0 \dots \mathbf{p}_{N+5}$ where two corners are ordinary and one corner has valence N. We can bound the pseudo-normal patch (and thus the Gauss map) of this limit patch.

Consider $\hat{\mathbf{x}}$, a planar approximation of $\mathbf{x}$. A particularly suitable approximation is derived by taking the first three terms of the eigenbasis expansion around the extraordinary vertex:

$$\begin{aligned}\hat{\mathbf{x}}(u,v) &= \sum_{i=0}^{2} \mathbf{q}_i \Phi^i(u,v) \\ \mathbf{x}(u,v) &= \sum_{i=0}^{N+5} \mathbf{q}_i \Phi^i(u,v) = \hat{\mathbf{x}}(u,v) + \textit{h.o.t.}\end{aligned}$$

where $\mathbf{L}_i$ are the left eigenvectors of the eigenspace and $\mathbf{q}_i = \mathbf{L}_i \mathbf{p}$.

Because it is based on the characteristic map, this approximation exhibits rapid convergence as subdivisions are performed. The partials of the original surface are

$$\mathbf{x}_u(u,v) = \mathbf{q}_1 \Phi_u^1 + \sum_{i=3}^{N+5} \mathbf{q}_i \Phi_u^i(u,v)$$

$$\mathbf{x}_v(u,v) = \mathbf{q}_2 \Phi_v^2 + \sum_{i=3}^{N+5} \mathbf{q}_i \Phi_v^i(u,v)$$

These partials are bounded by

$$\Box\mathbf{x}_u = \mathbf{q}_1 \Phi_u^1 + \sum_{i=3}^{N+5} \mathbf{q}_i \Box\Phi_u^i, \quad \Box\mathbf{x}_v = \mathbf{q}_2 \Phi_v^2 + \sum_{i=3}^{N+5} \mathbf{q}_i \Box\Phi_v^i.$$

The bound on the pseudo-normal patch of x (and the associated Gauss map) follows directly:

$$\Box N = \Box\mathbf{x}_u \times \Box\mathbf{x}_v.$$

Implementation $\Box\Phi_u^i$, $\Box\Phi_v^i$ and L_i depend on the connectivity and edge tags, but not on the geometry of the control mesh. Thus they are precomputed. The bounds $\Box\Phi_u^i$ and $\Box\Phi_v^i$ are computed by finite differences on a (sufficiently refined) approximating mesh.

3.3 Removing the Collisions

When we detect a collision in the transition from time t_n to t_{n+1}, we examine the vertices v_i of all triangles involved in the collision. We assign parameters $s_i \in [0,1]$ that linearly interpolate the displacement over the time step. The goal is to find values for all the s_i that satisfy all collision constraints while maximizing an objective function. The objective function may be based on physical laws, such as conservation of momentum or energy.

In the current implementation, the objective is to position the vertices as close as possible to the position originally predicted by the integrator. Due to the coupling of the variables, a naïve binary search technique will not work. Nevertheless, the problem permits an easy solution. Initially we set $s_i = 0$ for all vertices, so that the system is free of interference. Every iteration we add increments of ϵ to all the s_i in round-robin fashion. If advancing a vertex causes interference, the ϵ increment for that vertex is halved. Successful termination occurs when the increment for all vertices is below a threshold.

4 Finite Element Integral Evaluations

Equation 2 requires the evaluation of integrals involving derivatives of the Loop basis functions. Numerical quadratures are required since these non-linear expressions do not admit closed form solutions. In the context of finite element simulations we only need to ensure that the quadratures are computed to the necessary precision. In this section we describe a simple technique to evaluate the desired quantities at selected quadrature points at very low cost. Note that we are not approximating the evaluations but are evaluating the limit functions themselves.

Evaluation of limit surface quantities can be performed with the aid of an arbitrary parameter evaluation routine [26] for subdivision surfaces. Unfortunately the number of cases to be taken into account when allowing for tags is large and cumbersome to deal with. Since we only require evaluation at selected fixed parameter space points it is much more economical to specialize a priori for these cases. Candidate quadrature rules are either the centroid or edge midpoint rule (see Figure 6). The former was employed successfully in [2, 3] without the presence of tags. In the presence of tags the edge midpoint Gauss rule is simpler than the centroid rule. In both cases evaluation of the limit surface quantities is achieved by some number of (conceptual) subdivision steps until the selected quadrature point is entirely contained within a regular region of the control mesh. In the presence of tags this requires three levels of subdivision for the centroid Gauss quadrature, but only two levels for the edge midpoint Gauss rule, favoring the latter.

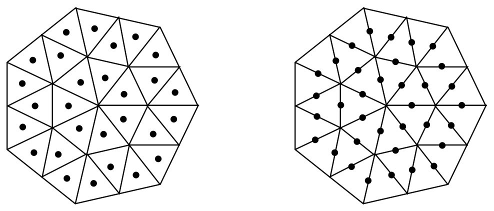
Figure 6: *Comparisons of centroid (left) and midedge (right) quadrature rules for a single basis function (here: the case of valence $k = 7$). The centroid rule uses $4k$, while the midedge rule uses $5k$ quadrature points per basis function.*

Having settled on the edge midpoint quadrature rule we must compute the Taylor series expansion of the limit surface up through second order at the limit position corresponding to the midpoint of an edge. Note that this is always well defined. Figure 7 illustrates the setup for an interior edge. In the presence of tags the edge may be influenced by the control points at the ends of all edges shown. Denote this set with the vector $\mathbf{p}^0$ of nodal positions. Performing two levels of subdivision for the interior ring only (see Figure 7 middle and right) results in a regular setting around the midpoint of the edge independent of any tags at the coarsest level. These seven nodal positions are given as $\mathbf{p}^2 = \mathbf{S}_2\mathbf{S}_1\mathbf{p}^0$, where $\mathbf{S}_2$ and $\mathbf{S}_1$ collect up all the subdivision masks to determine $\mathbf{p}^2$ from $\mathbf{p}^0$. The exact structure of $\mathbf{S}_2$ and $\mathbf{S}_1$ depends on any tags in the neighborhood shown. If we know the limit positions of $\mathbf{p}^2$ we may construct a second degree interpolating polynomial which is coincident with the Taylor series of the surface at the edge midpoint up to second order. This cannot be done directly since the limit positions of $\mathbf{p}^2$ depend also on neighbors which we did not compute. However, taking advantage of the fact that the left eigen vectors $\mathbf{L}$ associated with the eigen values $\{1, 1/2, 1/2, 1/4, 1/4, 1/4\}$ of the regular subdivision operator are only supported on the inner most ring we can compute the associated eigen coefficients, $\mathbf{L}\mathbf{p}^2$ with just the seven nodal positions we have. The associated right eigenvectors $\mathbf{R}$, are supported on a 2-ring. It is an easy matter to evaluate $\mathbf{R}$ on the limit surface for the 1-ring of interest. $\mathbf{R}^\infty$ may be found through application of the limit mask. Multiplying it with $\mathbf{L}\mathbf{p}^2$ we get the desired seven limit positions. Letting $\mathbf{P}$ denote the Vandermonde matrix on the inner ring up to order two we get the desired quantities in the center as

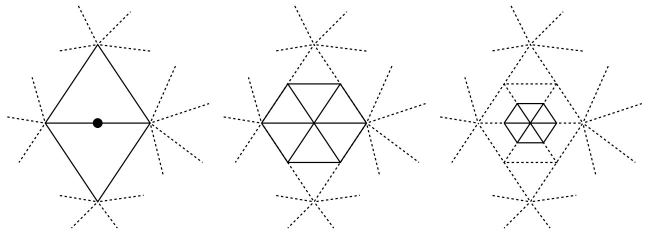
Figure 7: *For an arbitrary edge, possibly with tags in its vicinity two levels of subdivision around the middle of the edge results in a completely regular 1-ring (right) for which simple limit analysis is available.*

$$\begin{pmatrix} 12 & p \\ 3 & p_u \\ \sqrt{3} & p_v \\ & p_{uu} \\ \sqrt{3} & p_{uv} \\ 3 & p_{vv} \end{pmatrix}^\infty = (\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T\mathbf{R}^\infty\mathbf{L}\mathbf{p}^2 = \mathbf{M}\mathbf{p}^2$$

$$= \begin{pmatrix} 6 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 4 & 2 & -2 & -4 & -2 & 2 \\ 0 & 0 & 2 & 2 & 0 & -2 & -2 \\ -32 & 16 & 0 & 0 & 16 & 0 & 0 \\ 0 & 0 & 16 & -16 & 0 & 16 & -16 \\ -96 & -16 & 32 & 32 & -16 & 32 & 32 \end{pmatrix} \mathbf{p}^2,$$

where we moved some normalization constants to the left hand side for clarity. The first column of $\mathbf{M}$ corresponds to the center vertex and the following columns enumerate the 1-ring neighbors in counterclockwise order starting at three o'clock. One-sided stencils on the boundary follow in the usual way through reflection along the boundary edge. The assumed parameterization aligns the u- and v-axis parallel to (resp. perpendicular to) the unit edge incident on two equilateral triangles. The actual control mesh triangle shapes are accounted for through an appropriate affine transformation which enters as a change of variable into the desired integrals.

To arrive at the final dependence on $\mathbf{p}^0$ we apply two levels of adjoint subdivision to the rows of $\mathbf{M}$. This can be done easily in the presence of tags with a few staticly sized arrays. As a result the implementation does not need to contain the usual quadtrees or other data-structures to support subdivision. All that is needed is the coarsest level mesh and a relatively small piece of code to apply $\mathbf{S}_1^T\mathbf{S}_2^T$.[1]

5 Results

We have implemented the described methods and computed a series of examples which we briefly discuss here.

Crushing Can The first example is the crushing of a very thin cylinder. During the crushing process the displacements at the bottom of the cylinder are fixed and prescribed at the top. We linearly increase the top displacements as a function of time. Although not indicated in the Figures the cylinder has rigid diaphragms at both ends. The cylinder buckles already in the initial stage of the simulation and exhibits the deformation pattern as shown in Figures 8 and 1 (leftmost). This type of structural behavior can of course only be captured with nonlinear mechanics.

[1]The code for this can be found on the proceedings CDROM.

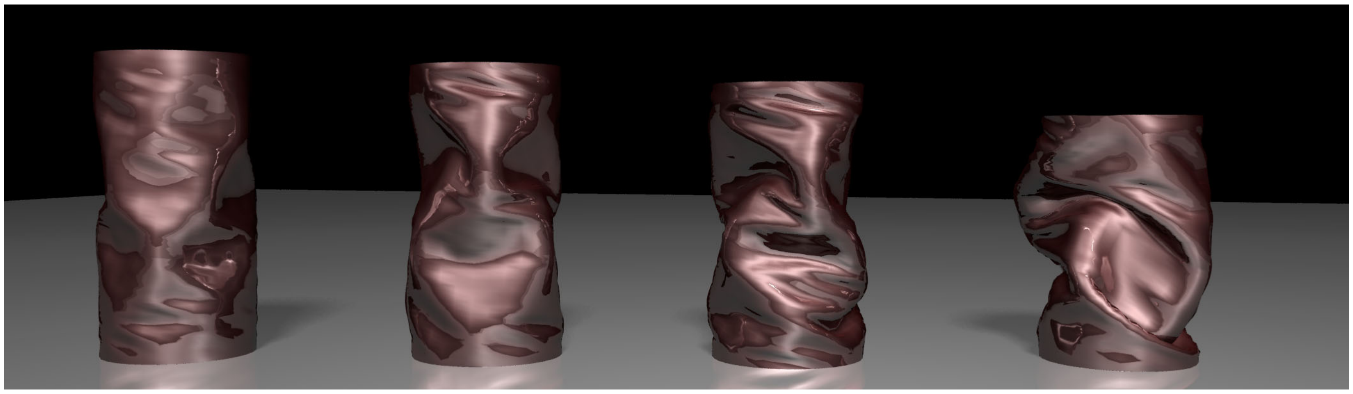

Figure 8: *Crushing a can.*

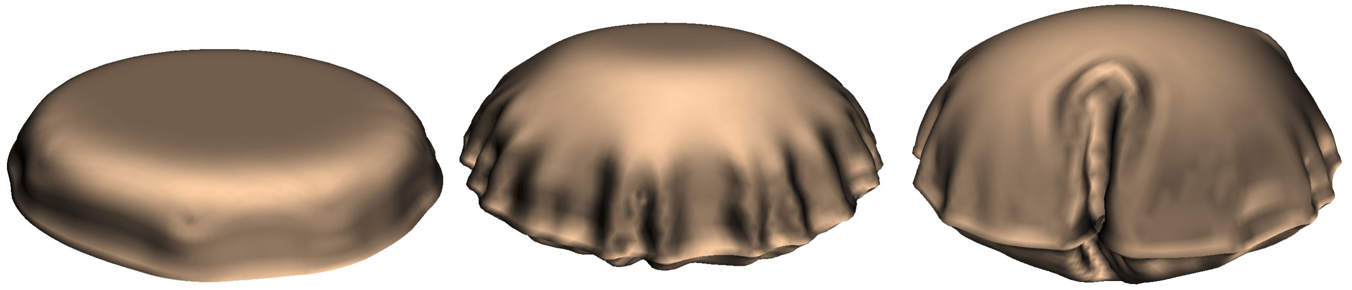

Figure 9: *Three time steps during the simulation of an inflating airbag.*

Inflating Airbag The second example is the inflation of an airbag. The airbag geometry in the undeformed configuration consists of two flat pieces attached to each other at the boundaries. In contrast to the cylinder example the mesh has irregular vertices. In order to inflate the airbag we apply internal pressure. The inflation of the airbag leads to complex wrinkling patterns as shown in Figure 9.

6 Conclusion

We described a succinct and concise treatment of thin-shell equations which is fully scalable in the sense that different material models, collision response methods, or integrators can easily be included. In particular this makes it possible to cover a range of requirements from quick animation previews to full blown engineering design studies. The use of subdivision surfaces as the basic primitive for both geometry and finite element treatments is very attractive. We discussed the collision detection and response problem and computed a number of examples to demonstrate the method.

So far many applications of physical simulation in computer graphics have been limited to linear models. These are not sufficient to exhibit the subtle and beautiful behaviors we find in real world objects. Compare the photograph in Figure 8 with the pictures from our simulations. We argue that there is little reason not to include these effects.

References

[1] ANONYMOUS. Curvature Smoothness of Subdivision Surfaces. submitted for publication, 1999.

[2] ANONYMOUS. Subdivision Surfaces: A New Paradigm for Thin-Shell Finite-Element Analysis. *Int. J. Numer. Meth. Engng. 47* (2000).

[3] ANONYMOUS. Subdivision Surfaces: Large Deformation Analysis of Elastic Thin-Shells. submitted for publication, 2000.

[4] BENSON, D. J., AND HALLQUIST, J. O. A Single Surface Contact Algorithm for the Post-Buckling Analysis of Shell Structures. *Computer Methods in Applied Mechanics and Engineering* (1990).

[5] BIERMANN, H., LEVIN, A., AND ZORIN, D. Smooth Subdivision Surfaces with Normal Control. Tech. rep., Courant Institute of Mathematical Sciences, New York, 1999.

[6] CARMO, M. P. D. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976.

[7] CATMULL, E., AND CLARK, J. Recursively Generated B-Spline Surfaces on Arbitrary Topological Meshes. *Computer Aided Design 10* (1978), 350–355.

[8] CELNIKER, G., AND GOSSARD, D. Deformable Curve and Surface Finite Elements for Free-Form Shape Design. *Computer Graphics (Proceedings of SIGGRAPH 91) 25*, 4 (1991), 257–266.

[9] COHEN, J. D., LIN, M. C., MANOCHA, D., AND PONAMGI, M. K. I-COLLIDE: An Interactive and Exact Collision Detection System for Large-Scale Environments. In *Proc. ACM Interactive 3D Graphics Conf.*, 189–196, 1995.

[10] DOO, D., AND SABIN, M. Behaviour of Recursive Division Surfaces Near Extraordinary Points. *Computer Aided Design 10* (1978), 356–360.

[11] GREINER, G. Variational Design and Fairing of Spline Surfaces. *Computer Graphics Forum 13*, 3 (1994), 143—154.

[12] HALSTEAD, M., KASS, M., AND DEROSE, T. Efficient, Fair Interpolation Using Catmull-Clark Surfaces. *Proceedings of SIGGRAPH 93* (1993), 35–44.

[13] HOHMEYER, M. A surface intersection algorithm based on loop detection. *International Journal of Computational Geometry and Applications* (1991).

[14] HUGHES, M., DIMATTIA, C., LIN, M. C., AND MANOCHA, D. Efficient and Accurate Interference Detection for Polynomial Deformation. *Proceedings of Computer Animation '96* (1996).

[15] HUGHES, T. *The Finite Element Method*. Prentice-Hall Inc., 1987.

[16] KANE, C., REPETTO, E., ORTIZ, M., AND MARSDEN, J. Finite Element Analysis of Nonsmooth Contact. *Comp. Meth. Appl. Mech. Engrg. 180* (1999).

[17] LEIF P. KOBBELT, K. DAUBERT, H.-P. S. Ray Tracing of Subdivision Surfaces. *Eurographics Rendering Workshop Proceedings* (1998).

[18] LI, T.-Y., AND CHEN, J.-S. Incremental 3D Collision Detection with Hierarchical Data Structures. *ACM Symposium on Virtual Reality Technology* (1998).

[19] LIN, M., AND GOTTSCHALK, S. Collision Detection between Geometric Models: A Survey. *Proceedings of IMA Conference on Mathematics of Surfaces* (1998).

[20] LOOP, C. Smooth Subdivision Surfaces Based on Triangles. Master's thesis, University of Utah, Department of Mathematics, 1987.

[21] MANDAL, C., QIN, H., AND VEMURI, B. C. A Novel FEM-Based Dynamic Framework for Subdivision Surfaces. In *Proceedings ACM Solid Modeling '99*, 191–201, 1999.

[22] QIN, H., MANDAL, C., AND VEMURI, B. C. Dynamic Catmull-Clark Subdivision Surfaces. *IEEE Transactions on Visualization and Computer Graphics 4*, 3 (1998), 215–229.

[23] QIN, H., AND TERZOPOULOS, D. D-NURBS: A Physics-Based Framework for Geometric Design. *IEEE Transactions on Visualization and Computer Graphics 2*, 1 (1996), 85–96.

[24] QIN, H., AND TERZOPOULOS, D. Triangular NURBS and their dynamic generalizations. *Computer Aided Geometric Design 14*, 4 (1997), 325–347.

[25] SIMO, J., AND HUGHES, T. *Computational Inelasticity*. Springer-Verlag, 1998.

[26] STAM, J. Exact Evaluation of Catmull-Clark Subdivision Surfaces at Arbitrary Parameter Values. *Proceedings of SIGGRAPH 98* (1998), 395–404.

[27] TERZOPOULOS, D., AND QIN, H. Dynamic NURBS with Geometric Constraints for Interactive Sculpting. *ACM Transactions on Graphics 13*, 2 (1994), 103–136.

[28] TIMOSHENKO, S., AND WOINOWSKY-KRIEGER, S. *Theory of Plates and Shells*. McGraw-Hill Book Company Inc., 1959.

[29] VOLINO, P., COURCHESNE, M., AND THALMANN, N. M. Versatile and Efficient Techniques for Simulating Cloth and Other Deformable Objects. In *Proc. SIGGRAPH'95*, 137–144, 1995.

[30] VOLINO, P., AND THALMANN, N. M. Efficient Self-Collision Detection on Smoothly Discretised Surface Animations Using Geometrical Shape Regularity. In *Proc. EuroGraphics'94*, 155–166, 1994.

[31] WELCH, W., AND WITKIN, A. Variational surface modeling. *Computer Graphics (Proceedings of SIGGRAPH 92) 26*, 2 (1992), 157–166.

[32] ZORIN, D., AND SCHRÖDER, P., Eds. *Subdivision for Modeling and Animation*. Course Notes. ACM SIGGRAPH, 1999.