

# Embedded Thin Shells

for Wrinkle Simulation

Olivier Remillard

Master of Science

School of Computer Science

McGill University

Montreal, Quebec

March 2013

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of master of science.

©Olivier Remillard 2013

## ACKNOWLEDGEMENTS

The author wishes to express his gratitude to his supervisor, Paul G. Kry who was abundantly helpful and offered invaluable assistance, support and guidance. Special thanks also to members of the computer graphics lab, especially to Sheldon Andrews for sharing his collision detection code.

## 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.

## ABRÉGÉ

Nous présentons une nouvelle technique pour la simulation des déformations de haute résolution du plissement en surface d'objets composites constitués d'un intérieur mou et d'une peau plus rigide. Nous combinons des plaques minces haute résolution avec une structure d'éléments finis plus approximative, et définissons les contraintes d'accouplement qui permettent la formation de rides qui possèdent des propriétés correspondant celles prédites par les paramètres physiques de l'objet composite. Le couplage de ces deux modèles permet notre procédé de produire le comportement attendu sans la charge de calcul d'un grand nombre d'éléments volumiques du aux déformations sous la surface du modèle. Nous utilisons des fonctions de forme quadratiques B-splines pour les déformations intérieures, ce qui permet des résolutions très grossières pour modéliser les déformations globales efficacement, tout en évitant les artefacts visuels de ridules au niveau des limites de discrétisation. Nous démontrons que ce modèle produit des longueurs d'onde de rides qui correspondent aux prédictions théoriques et aux simulations volumétriques de haute résolution. Nous présentons aussi des exemples d'applications en simulant le plissement d'objets passifs, tels que des meubles, et aussi de la formation de rides sur le visage de personnages en animation.

## TABLE OF CONTENTS

ACKNOWLEDGEMENTS		ii
ABSTRACT		iii
ABRÉGÉ		iv
LIST OF TABLES		vii
LIST OF FIGURES		viii
1	Introduction	1
2	Related Work	4
3	Coupling Surface and Interior	8
	3.1 Constraints with Local Support	10
	3.2 Global Spectral Constraints	11
4	Wrinkle Parameters	14
	4.1 Selecting Parameters	15
5	Model Composition	17
	5.1 Surface elements implementation	20
	5.2 Interior elements implementation	23
	5.3 Interior Construction	28
6	Simulation	30
	6.1 Surface Statics	31
	6.2 Surface Statics: A Revised Approach	33
	6.3 Interior Dynamics	37
7	Discussion	40
	7.1 Timings	40
	7.2 Full Simulation Comparison	40
	7.3 Critical Wavelength Comparison	43
	7.4 Natural Phenomena	43
	7.5 Facial Wrinkles	45
	7.6 Limitations	46

8	Conclusion . . . . .	51
	References . . . . .	53

LIST OF TABLES

<u>Table</u>		<u>page</u>
7-1	Simulation Timings . . . . .	41

## LIST OF FIGURES

<u>Figure</u>	<u>page</u>
1-1 Wrinkles . . . . .	2
3-1 Surface Mesh Partitioning . . . . .	11
3-2 Spectral and Local Coupling . . . . .	13
5-1 Linear and C1 Coarse Elements under High Frequency Wrinkles	18
5-2 Linear and C1 Coarse Elements under Low Frequency Wrinkles	19
5-3 Shell Elements . . . . .	20
5-4 Quadratic B-spline elements . . . . .	24
5-5 Lattice Construction . . . . .	28
6-1 Constraint QR-Decomposition . . . . .	36
7-1 Compression Experiment . . . . .	42
7-2 Comparison of Embedded Thin Shells and Dense Volumetric Simulation . . . . .	43
7-3 Theoretical and Simulated Material Critical Wavelength . . . .	44
7-4 Natural Phenomena . . . . .	45
7-5 Face Wrinkles . . . . .	46
7-6 Animation Sequences with Different Material Properties . . . .	49
7-7 Animation Sequences with Different Large Deformations . . . .	50

## CHAPTER 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 [11]. The phenomenon is also observed in other examples of composite materials at various scales, from dried fruit to mountain formation [13].

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

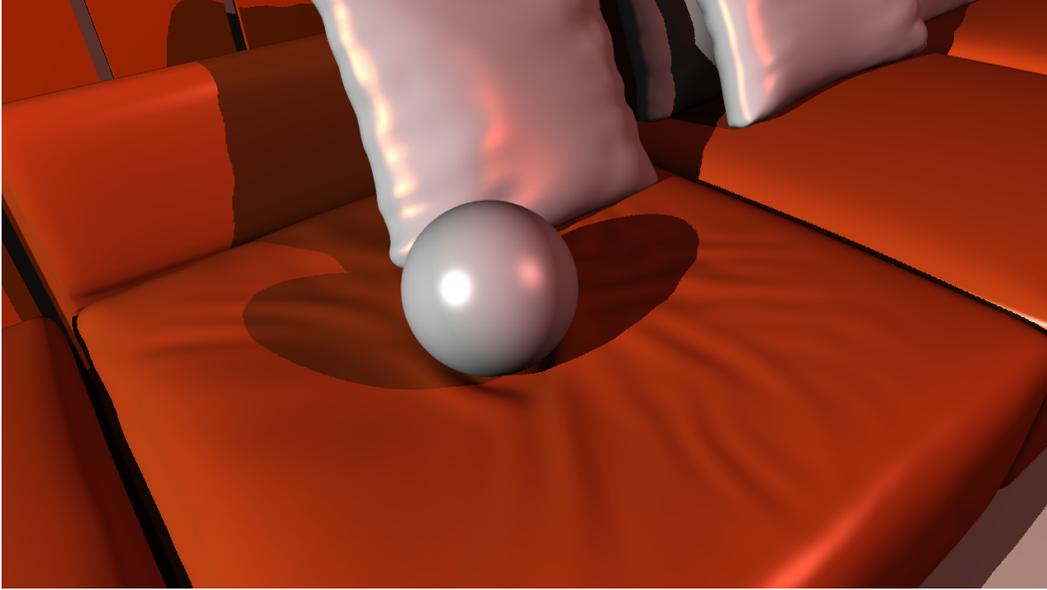


Figure 1–1: An example of wrinkles produced by our technique.

preferable to have a simulation, for example, in wrinkling due to arbitrary contacts with the environment.

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 finite element 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 finite element 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 1–1 shows a preview of the results.

We first acknowledge previous work on which the model is motivated and built upon in Chapter 2 Related Work. We offer the description of two ways for constructing the coupling constraint in Chapter 3 Coupling Surface and Interior. Chapter 4 Wrinkle Parameters continues the description by analyzing the effect of the material properties to the modeling parameters coming into the constrain creation. We then present in detail the thin shell and volume model we used in this work. We give light on their energy formulation, construction and implementation details throughout Chapter 5 Model Composition. We later explain the equations of motion and discuss the solvers coming into the simulation of the model in Chapter 6 Simulation. Timings and animation sequences are offered in Chapter 7 Discussion where we also provide examples and experiments to prove the validity of this model altogether with its limitations. We summarize the embedded thin shells model a final time and elaborate on future work in Chapter 8 Conclusion.

## CHAPTER 2 Related Work

Early seminal work on elastic deformable models introduced techniques based on finite elements to computer animation [41, 43]. In other early work, free form deformation lattices [36] were used to define specific deformation modes, for which physically based equations of motion could be created to produce animation [12]. 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 [9], corotational formulation for fast simulation with linear elasticity [27], fracture [28, 33], node duplication to deal with small pieces during fracture [26], and elements robust to inversion [15]. Wojtan and Turk [48] 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 [31]. 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 [42, 38]. 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. [22] 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 [24]. 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. [7] 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. [8] 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. [14], though alternatives exist, for instance a quadratic energy model for inextensible surfaces [3]. We have explored different options for stretching and shearing energy, but we ultimately use the model of Baraff and Witkin [1] 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 [44] and physically based shape editing [25]. 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 [2, 18]. 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. [2]. 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. [39]. For single point contact, Seiler et al. [37] 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 [35], or even designed as part of a character skin [20, 47]. Other work looks specifically at wrinkles and folds of bending fingers [46]. Certainly, artist designed wrinkle maps (or normal maps) are likely the most common technique for modeling and rendering fine details since the early days [4], and are still widely used in current applications [17]. 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.

### CHAPTER 3

#### Coupling Surface and Interior

In this chapter, we present the core concept behind embedded thin shells. We address the simulation of soft objects with harder skin by separating the surface wrinkling from the interior's deformation. Thereafter, the two models interact with each other and produce detailed wrinkle patterns and rich deformation. The constraint must be powerful enough to transmit interior's deformations to the surface while being lenient toward the surface wrinkling. We shed light on two frequency based constraints possessing these antipodal properties in the next sections.

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  $B : \mathbb{R}^m \rightarrow \mathbb{R}^n$  using interpolation weights of the surface mesh with respect to the hexahedral lattice given the mesh position at the reference configuration. The reference configuration should represent 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

$$x_{q_0} = Bq_0.$$

The 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 structure 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 finite element 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 Chapter 4, while the next two sections 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. [2] 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

$$\sum_{i \in C} \alpha_i x_i = \sum_{i \in C} \alpha_i [Bq]_i, \quad (3.1)$$

where the weights  $\alpha_i$  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,

$$\alpha_i = w_C e^{-\frac{d_i^2}{2\sigma^2}} \text{ if } d_i < 2\sigma, \quad 0 \text{ otherwise,} \quad (3.2)$$

where  $w_C$  normalizes the sum of weights to one, and  $d_i$  is the distance from vertex  $i$  to the cluster’s centre, which we approximate by the shortest path following mesh edges. The standard deviation  $\sigma$  determines the frequency attenuation power of this Gaussian as a spatial filter. The truncation at  $2\sigma$  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 centred at locations distributed evenly across the surface. The cluster centres partition the mesh into regions as seen in Figure 3–1. Centres are greedily selected in a preprocessing step using breadth first search along mesh edges. We start by choosing a random vertex for the first centre, and then compute the minimum edge traversal

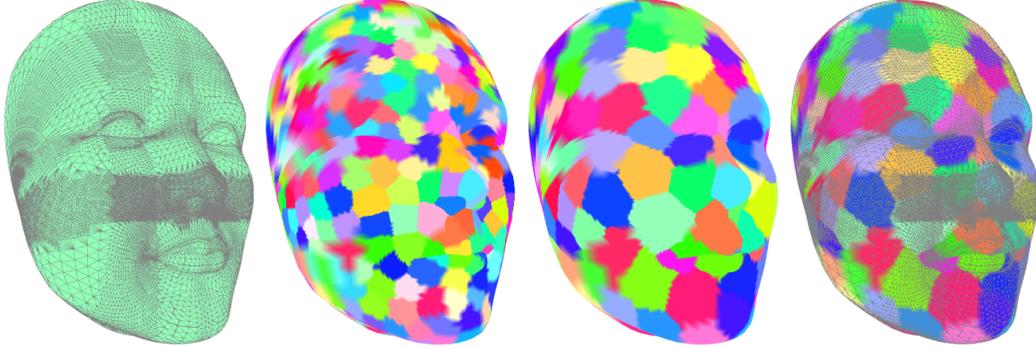


Figure 3–1: An example mesh with two different resolution partitions into uniform regions. Our greedy algorithm is unaffected by varying density in the mesh. Region centres define the locations for overlapping Gaussian weighted averages that make up our constraints.

distance to all other vertices. We choose the farthest vertex as the next centre, and then update the minimum distances for all the vertices that are closer to this centre. We repeat until all vertices have at most a distance of  $\frac{1}{2}r$  to a centre, where  $r$  is the desired distance between centres 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.

### 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. [18] 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 3–2.

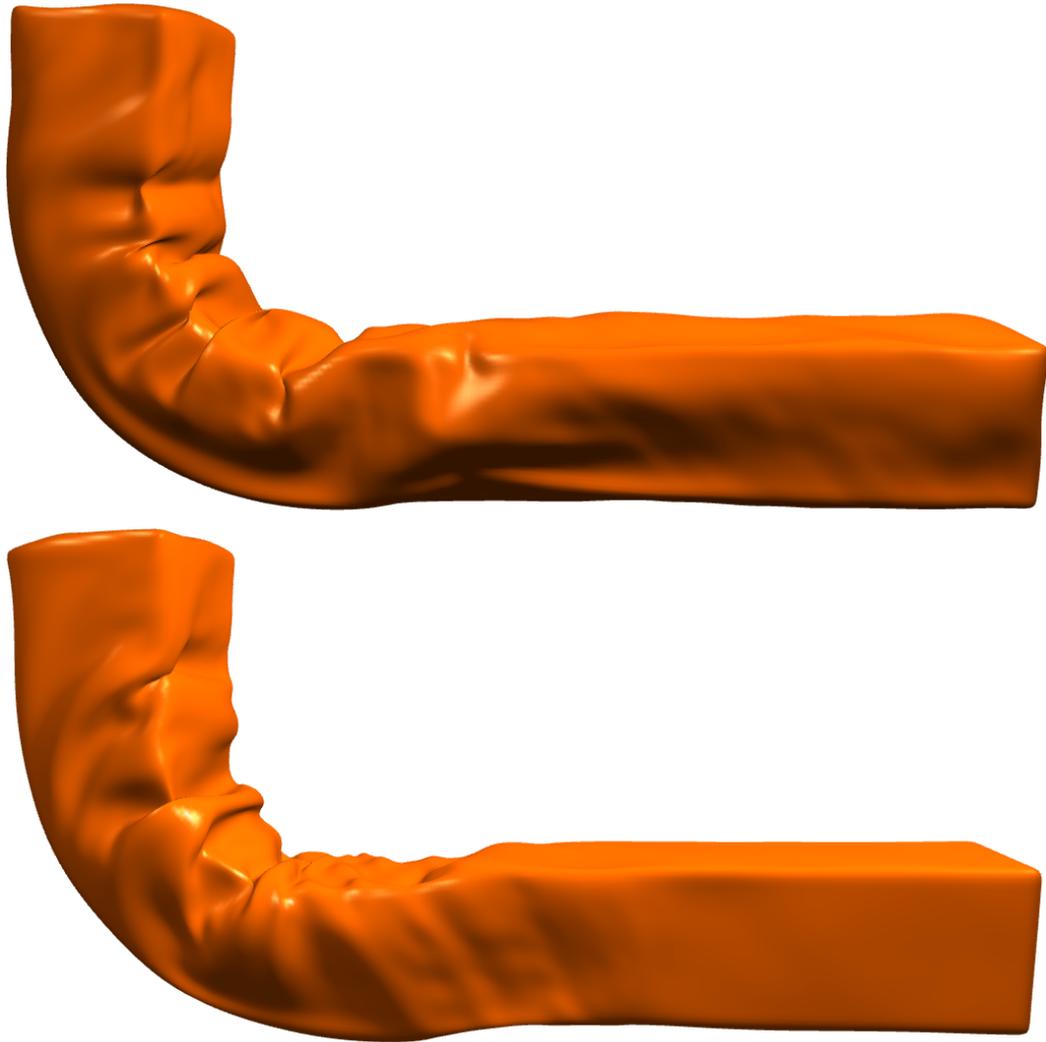


Figure 3–2: When bending a rectangular solid, spectral constraints produce ripples across the entire model (top), while local constraints allow wrinkling effects to remain localized (bottom).

## CHAPTER 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 behaviour of a skin attached to a soft elastic foundation [45]. 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  $\lambda$  along the compression direction. Given the thickness of the skin  $h$ , and the Young's modulus and the Poisson's ratio of the skin and the interior ( $E_x, \nu_x$ , and  $E_q, \nu_q$  respectively), the corresponding critical wavelength is

$$\lambda = 2\pi h \left[ \frac{(1 - \nu_q^2)E_x}{3(1 - \nu_x^2)E_q} \right]^{1/3}. \quad (4.1)$$

Using the critical wrinkling wavelength for our given object's parameters, we can compute the inter-cluster spacing  $r$  and the standard deviation  $\sigma$  to use in the construction of our constraints.

Recall that the averaging weights in a given cluster come from a Gaussian with the shape

$$g(d) = e^{-\frac{d_i^2}{2\sigma^2}}. \quad (4.2)$$

The parameter  $\sigma$  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

$$G(\xi) = \sigma \sqrt{\frac{\pi}{2}} e^{-2\sigma^2 \pi^2 \xi^2}, \quad (4.3)$$

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

$$\frac{1}{\lambda} = 2(2\sigma\pi)^{-1}, \quad \text{thus,} \quad \sigma = \lambda/\pi. \quad (4.4)$$

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 4.5 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 4.5.

We note that there are alternatives to our greedy approach to building well-spaced cluster centres for local constraints, and examples are discussed by Bergou et al. [2]. 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 centres 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 of  $h = 2$  mm and Young's modulus  $E_x = 40$  MPa, and foam interior with  $E_q = 0.5$  MPa, giving wrinkles of wavelength approximately 4 cm.

The stiffness parameters for the elastic stretching and bending energy of the surface are set as  $hE_x$  and  $h^3E_x$  respectively to conform to the model parameters. We note that the Poisson's ratio does not play a prominent part in determining the wavelength and can largely be ignored, even when the surface and interior have varying Poisson's ratios

$$0.9 \times 2\pi h \left[ \frac{E_x}{3E_q} \right]^{1/3} < \lambda < 1.1 \times 2\pi h \left[ \frac{E_x}{3E_q} \right]^{1/3}. \quad (4.5)$$

Where the longest wavelengths are obtained when the Poisson's ratio of the interior is at  $\nu_q = 0.5$  and the surface ratio is zero. The reverse relationship produces the shortest wavelengths: the surface ratio at maximum and the interior to zero.

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.

## CHAPTER 5

### Model Composition

We use dense triangle meshes with shell stretch and shear energy defined by Baraff and Witkin [1], and bending energy defined by Grinspun et al. [14]. Thin shell energies are best suited to regular meshes, and have poor behaviour 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 [21].

We use invertable rotated linear energy for our hexahedral lattice [16], 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 based on coarse discretization will form at element boundaries due to pinching or discontinuous surface tangents. This artifact is absent with our quadratic shape functions because they are  $C^1$  continuous. Figures 5–1 and 5–2 show a demonstration of the artifacts that are produced when using linear shape functions in the interior.

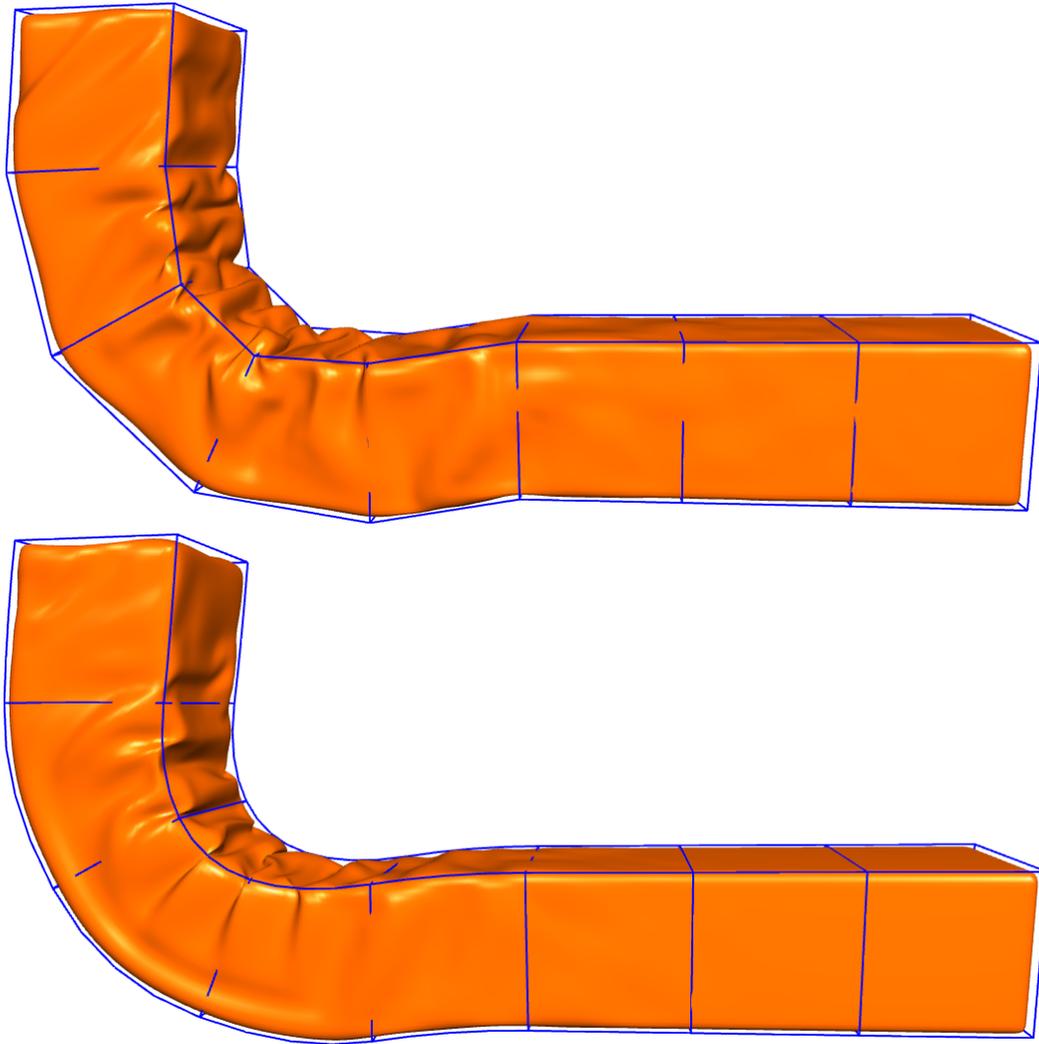


Figure 5–1: A comparison of linear and quadratic shape functions for interior elements showing that linear shape functions produce artifacts at high wrinkle frequencies where the  $C^0$  continuity at interior element boundaries are evident.

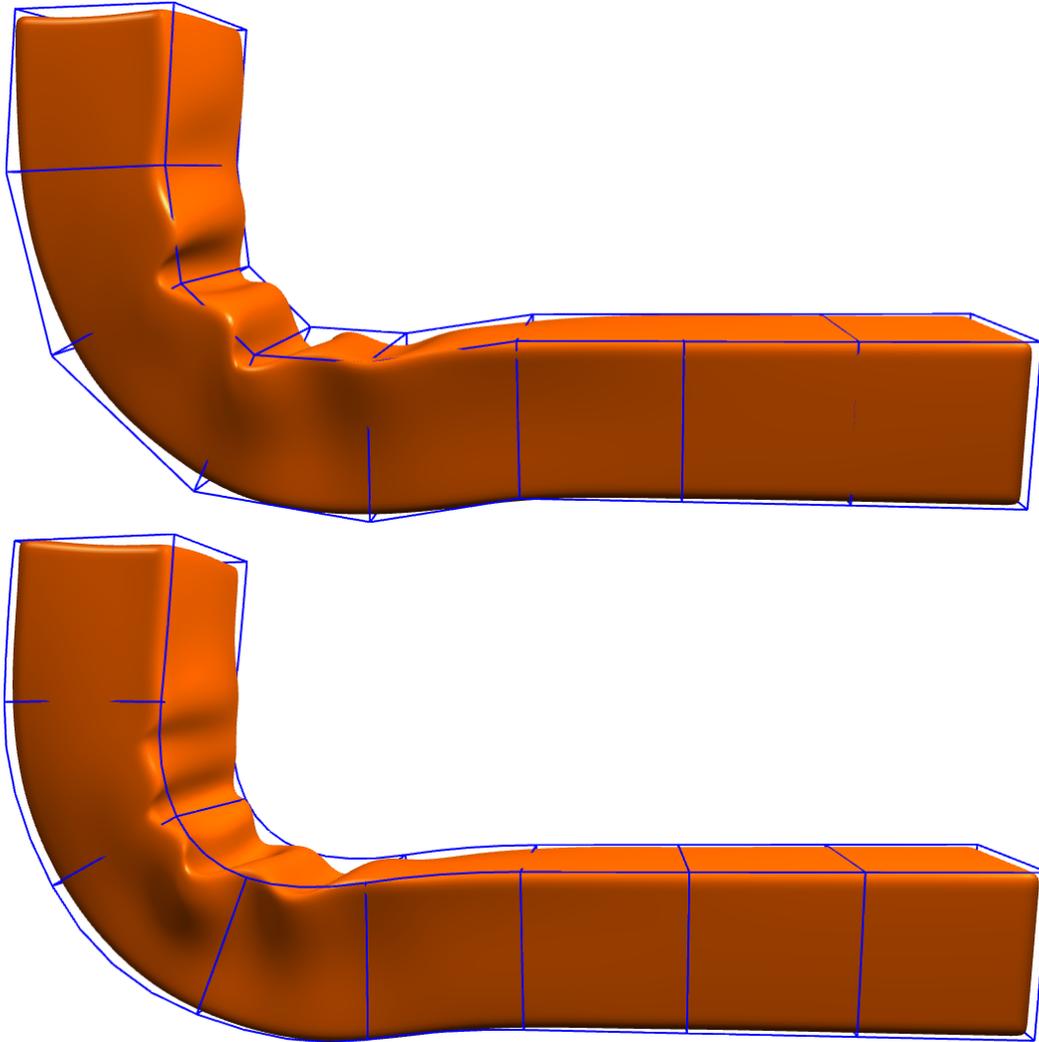


Figure 5–2: A comparison of linear and quadratic shape functions for interior elements showing that linear shape functions produce artifacts at low wrinkle frequencies where wrinkles creases fall at element boundaries, but otherwise the surface preserves its smooth aspect and element boundaries are not apparent from the surface.

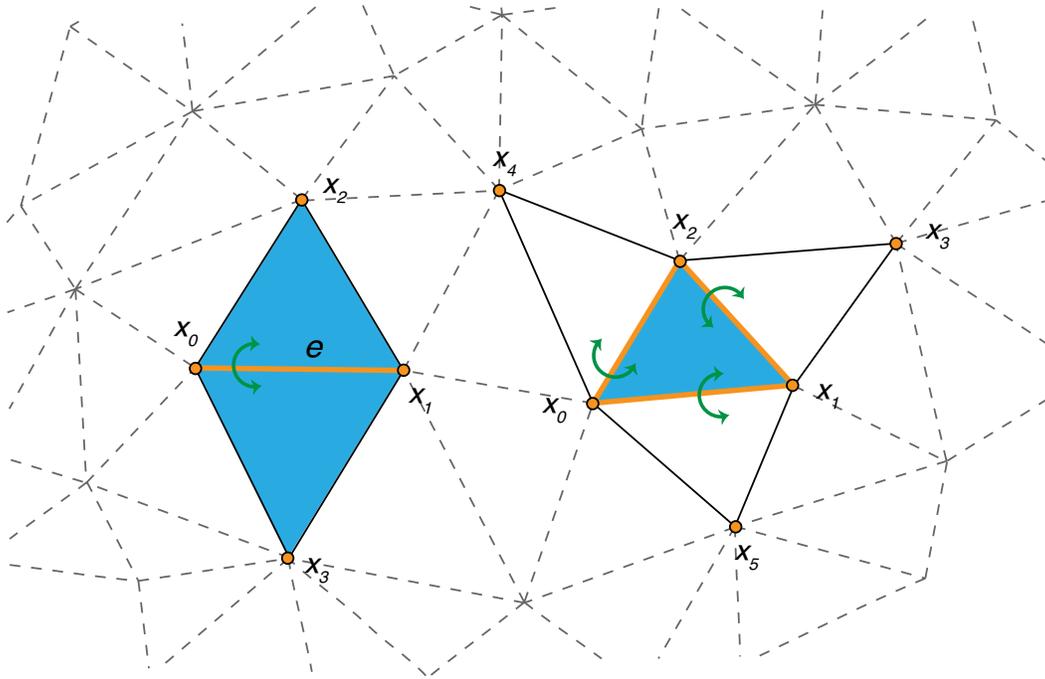


Figure 5–3: Edge element on the left and triangle element of the right. The blue triangle represents the triangle for which the stretch and shear energy must be computed, and the green arrows represents the bending energies. The edge element depends on 4 nodes while a triangle element requires 6 nodes.

### 5.1 Surface elements implementation

The object surface is discretized into a fine triangle mesh represented by finite elements in order to compute the stretching, shearing and bending energy of this surface.

We use the mesh face edges to symbolize the finite elements instead of the mesh triangle faces itself. By relying on the edges, the computation of the different energies acting upon the elements depend on fewer degrees of freedom. The bending energy operates on the angle between two adjacent triangle of the mesh. The stretch and shearing energy pertain simply to one triangle face configuration.

Thus, a concise way to express these energies is by utilizing the edge as an element. We illustrate the two types of elements in Figure 5–3. The

bending energy is described with the four nodes forming the two triangles along the element's edge and, concurrently, the stretching and shearing of the two attached faces is available. On the other hand, faces as elements force us to include the three nodes of the triangle face and another three nodes necessary to compute the bending energy of the three edges forming that face.

Both approaches suffer from redundant computations. The former computes three times the face stretch and shear of each face when looping over all edges, while the latter will duplicate the bending energy of each edge. Keeping this in mind, we simply multiply the stretching and shearing energy by a factor of a third or, to avoid computation on CPU, we devised a scheme to compute it only once. We solely compute the energy when the opposed nodes  $x_2$  or  $x_3$  from the edge has a lower global index from  $x_0$  and  $x_1$ . For all its worth, this optimization is futile on a GPU because the units skipping the computation would have to wait for the others to finish the task. We suggest using the scaling strategy when the computation is distributed on the graphics card.

For a given edge element, we order the edge node first as  $x_0$  and  $x_1$  followed by the two opposed nodes  $x_2$  and  $x_3$ . We repeat the same node global index in  $x_2$  and  $x_3$  for border edge as these are only connected to one face. When  $x_2 = x_3$  we avoid computing the bending energy for that edge and avoid computing the face energies twice. In general, we compute the face energies for the two adjacent faces.

For the triangle  $x_0, x_1$  and  $x_2$ , we describe the triangle face positions in space using the linear function  $T(u, v)$ .

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = T \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

The stretch along both axis is quantified using  $T_u = \delta T / \delta u$  and  $T_v = \delta T / \delta v$ .

Since we assume the displacement of the triangle to be linear, we have

$\Delta x = T_u \Delta u + T_v \Delta v$ . Solving for the stretch,

$$\begin{bmatrix} T_u & T_v \end{bmatrix} = \begin{bmatrix} x_0 & x_1 & x_2 \end{bmatrix} \begin{bmatrix} -1 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \|x_1 - x_0\| & \frac{|(x_1 - x_0)^T (x_2 - x_0)|}{\|x_1 - x_0\|} \\ 0 & \frac{|(x_1 - x_0) \times (x_2 - x_0)|}{\|x_1 - x_0\|} \end{bmatrix}^{-1}$$

where the rightmost matrix values are taken in the reference configuration.

The three elements of this inverse are store and used to speed up the computation of the face stretch, which leads to the expression of the surface stretching energy  $W_a$  and  $W_s$

$$W_a = \frac{k_a}{2} A^2 [(\|T_u\| - 1)^2 + (\|T_v\| - 1)^2]$$

$$W_s = \frac{k_s}{2} A^2 (T_u^T T_v)^2$$

where  $A$  is the area of the triangle reference configuration. The expression for the triangle  $x_0, x_3$  and  $x_1$  is deducted similarly.

The bending energy of the edge

$$W_b = \frac{k_b}{2} A^2 \frac{3 \|e\|}{\|e_1\| + \|e_2\|} (\text{atan2}(n_1 \times n_2, n_1^T n_2) - \theta_0)^2$$

where  $\theta_0$  is the angle between the two faces in the reference configuration,  $n_1$  and  $n_2$  the two triangle face normals.

The internal force and stiffness matrix of an element are obtained by computing the Jacobian and the Hessian of those energies. The process is straightforward and fast. The use of this type of element as discretization of the surface encourages parallel computation. There are thousands of them, their memory footprint is small and their computation simple.

## 5.2 Interior elements implementation

The volume continuum of the interior, as with the surface, is implemented using the finite element methodology. We segment the interior into a coarse structure of hexahedra elements to simplify the computation of the elastic energy of the continuum they represents.

We use isoparametric elements in order to interpolate nodal values in a generic matter, regardless of the complexity of the shape functions. The position of the continuum throughout the element is described as

$$q(r) = \sum q_i N_i(r) \tag{5.1}$$

where  $q(r)$  is the parametrized position of a particle at  $r \in [-1, 1]^3$  in the idealized hexahedron element,  $q_i$  the  $i$ th nodal position and  $N_i(r)$  the  $i$ th shape function of the element.

We work with quadratic B-spline as shape function. Like the traditional quadratic hexahedron element, the element is defined over 27 nodes instead of the 8 nodes for its linear analogue but does not have any of its node coinciding with its corners. As shown in Figure 5–4, one node is placed at the centre of the element and the 26 others are distributed at the centre of the adjacent elements. This sparser distribution of node makes the B-spline shape function an attractive alternative to linear element as the ratio of nodes per element necessary to describe the continuum is nearly the same.

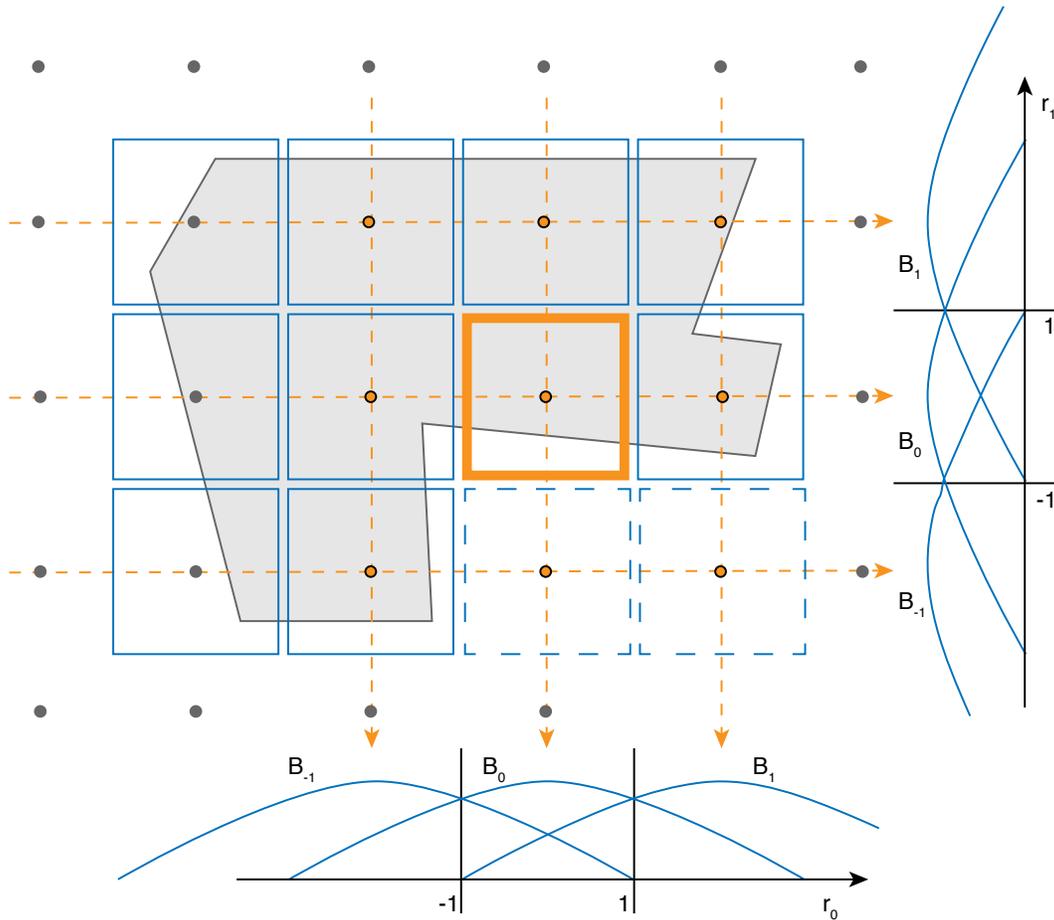


Figure 5–4: Simplified volume finite element model with embedded mesh. All the degrees of freedom are shown. We exposed the local frame of the orange element. There is one node at the centre of the element and the others all falls where the centre of the adjacent neighbours would be. In 2D, the element depends on 9 nodes, but in 3D, it depends on 27 nodes. The shape functions are displayed on the side of the element. The functions are valid in the range of  $r_0, r_1 \in [-1, 1]$ .

We produce the final shape function from the blend matrix  $B$

$$N_{i+2+3(j+1)+9(k+1)}(r) = B_i(r_0)B_j(r_1)B_k(r_2) \quad (5.2)$$

for  $i, j, k = -1, 0, 1$ . The function holds its name from the uniform quadratic B-Spline segment  $S_i(t)$  blending function defined over  $t \in [-1, 1]$  it uses as basis.

$$S_i(t) = B(t) \begin{bmatrix} q_{i-1} \\ q_i \\ q_{i+1} \end{bmatrix} = \frac{1}{8} \begin{bmatrix} t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 0 & 2 \\ 1 & 6 & 1 \end{bmatrix} \begin{bmatrix} q_{i-1} \\ q_i \\ q_{i+1} \end{bmatrix} \quad (5.3)$$

We measure the deformation between the reference configuration  $Q$  and the current state  $q$  of an element by analyzing its deformation gradient  $F = \frac{\partial q}{\partial Q}$ ,

$$F = \frac{\partial q}{\partial r} \left( \frac{\partial Q}{\partial r} \right)^{-1} = \sum q_i \frac{\partial N_i(r)}{\partial r} \left( \sum Q_i \frac{\partial N_i(r)}{\partial r} \right). \quad (5.4)$$

Because we base the construction of the continuum elements on a regular lattice, the gradient  $\frac{\partial Q}{\partial r}$  simplifies to  $\frac{h}{2}I$  where  $h$  is the dimension of the element. Once the deformation gradient is obtained, it can be used to construct the Green strain; yet, we diverge from this path and follow the work of Irving et al. [16] and then made more robust by Teran et al. [40]. They treat the elastic potential of the continuum using a constitutive model explicitly written with respect to the deformation gradient  $F$  instead of the common Green strain tensor. We avoid the Green strain for its invariance to all orthogonal transformations, including reflexions; therefore rendering Green strain tensor unsuited for correctly handling element inversion. Without proper treatment, an inverted element does not naturally recover as the elastic forces obtained through the orthogonal invariant stain density

function is oblivious to the inversion and yields forces that maintain the inversion.

In this setting, the elastic potential of an element is  $U_e = \int_V W(F)dV$  where  $W(F)$  is the strain density function. We obtain an equation for the elastic force for an element by taking the energy gradient

$$f^T = -\frac{\partial U_e}{\partial q} \quad (5.5)$$

$$\frac{\partial U_e}{\partial q} = \int_V \frac{\partial W(F)}{\partial q} dV = \int_V \frac{\partial W(F)}{\partial F} \frac{\partial F}{\partial q} dV = \int_V P \frac{\partial F}{\partial q} dV \quad (5.6)$$

where  $f$  is the elastic force and  $P$  the first Piola Kirchoff stress tensor. We approximate the integral by sampling the element.

$$\int_V P \frac{\partial F}{\partial q} dV = \int_V P \frac{h}{2} \frac{\partial^2 F}{\partial q^2} dV = \sum_g P|_g \frac{h}{2} \frac{\partial^2 F}{\partial q^2} \Big|_g \quad (5.7)$$

where the sample points are chosen according to Gauss quadrature sampling rules. Because we are using a higher order shape function, we based the approximation on three samples per dimension, for a total of 27 quadrature points.

The strategy of Irving and Teran is to use the stress tensor's invariance to the rotation for both the reference and the deformed space to decompose the tensor in a product of diagonal and rotation matrices. We express the stress tensor in the frame of principal direction of the deformation tensor  $F$ ,

$$P(F) = P(U\hat{F}V^T) = UP(\hat{F})V^T = \hat{P} \quad (5.8)$$

where  $U$  and  $V$  are the rotation and  $\hat{F}$  the deformation gradient diagonalization. This approach makes it easy to produce constitutive models  $P(F)$  for the element, where the relations is written in correspondence with the principal axis of the deformation. For this work, we use the rotated linear

model  $\hat{P} = 2\mu(\hat{F} - I) + \lambda tr(\hat{F} - I)$ . In this frame, inverted elements are corrected by making positive the smallest diagonal's entry of  $\hat{F}$ . This simple fix is sufficient to produce the necessary force vector that will recover the element from its inversion.

Contrary to the shell elements, we do not compute or assemble the element's stiffness matrix. We compute the force differential from stress differential obtained based on displacements. To compute the stress differential  $\partial P$ , we follow the algorithm of Teran and al. [40]. It is possible to do so even though we are using different shapes function than those from their work. The algorithm only depends on the use of the first Piola-Kirchoff stress tensor,

$$\partial P = U \left[ \frac{\partial P}{\partial F} \Big|_{\hat{F}} : U^T F V \right] V^T. \quad (5.9)$$

We compute the stress differential from the contraction of the deformation differential with  $\frac{\partial P}{\partial F}$ , the rotation  $U$  and  $V$  are the same used to rotate the deformation gradient to its principal axis. To assure the positive definitiveness of the element stiffness they show that it is sufficient to have  $\frac{\partial P}{\partial F}$  positive definite. Using the rotated linear constitutive model further simplifies this step. Its fourth order tensor becomes constant and positive definite when computed with respect to the diagonalized deformation gradient.

We make the process fast and parallelizable by using the fast SVD of 3x3 matrix described in the work of McAdams et al. [24]. By using an incomplete approximation of the square root, the singular value decomposition of the deformation gradient can be executed in a few nanoseconds. Furthermore, by fixing the number of iterations for the decomposition, it makes this process a candidate for parallelization on the GPU.

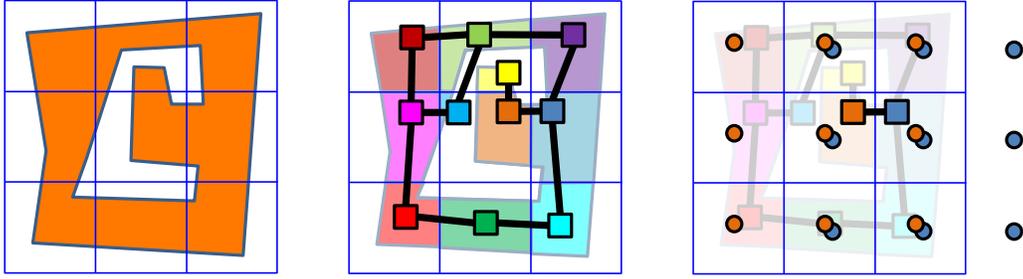


Figure 5–5: A 2D illustration of lattice construction. Left, the mesh is sliced along the lattice boundaries. Middle, the mesh segments are closed using the lattice cell faces, a finite element denoted by a coloured square is created for each, and connectivity across cell faces forms a graph. Right,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

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

The mass and stiffness properties of each boundary elements must be estimated since they are no longer fully enclosed in the volume but only represent a fraction of the object. We can scale their Young's modulus and redistribute the mass of the elements to its nodes based on the size and position of the enclosed volume. Also, the quadrature points can be optimized by repositioning them adequately as described by Patterson et al. [34]. Doing so, however, complicates the system by introducing elements with poor conditioning due to their small mass. The work of Nesme et al. [30] observes that as long as the stiffness is coherent with the mass, the system will remain stable.

## CHAPTER 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

$$F_x(x) = f_x - \left. \frac{\partial E_x}{\partial x} \right|_x^T, \quad (6.1)$$

$$F_q(q) = f_q - \left. \frac{\partial E_q}{\partial q} \right|_q^T. \quad (6.2)$$

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. [7].

### 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,

$$\text{solve}_{x,\lambda} \begin{cases} F_x + H^T \lambda = 0 \\ \gamma H(x_{qk} - x) = 0 \end{cases}, \quad (6.3)$$

and we introduce the scalar  $\gamma$  as a simple and inexpensive form of preconditioning to facilitate the solver progression. The value of  $\gamma$  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

$$\begin{bmatrix} \left. \frac{\partial^2 E_x}{\partial x^2} \right|_x & \gamma H^T \\ \gamma H & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \lambda \end{bmatrix} = \begin{bmatrix} -F_x(x) \\ \gamma H(x_{qk} - x) \end{bmatrix}. \quad (6.4)$$

We find that the solution of this system can be reliably computed with MINRES [32], while the method of conjugate gradients cannot be used for this system. Unfortunately, the surface energy Hessian is non-definite. As opposed to an implicit scheme where the Hessian is scale down and added lumped-mass matrix, there is no hope in making it positive definite.

We augment our constrained Newton-Raphson solver with a line search at each step using the Armijo rule. We search for step size  $\beta_m$  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  $\beta_m$  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  $\beta^*$  of the function  $\phi$ , 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  $\beta^*$  does not satisfy the criteria, with the model  $\phi$  reconstructed at each step using the last two  $\beta$  and the value at zero (see work by Kelly [19]).

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.

In effort to ameliorate the conditioning of the previous system, we reformulate the problem to allow for a different treatment of the constraint. The difficulty of solving the constraint with Lagrange multipliers is that no matter how we arrange or permute the system, the diagonal of the matrix will always contain zero or negative elements. One way to avoid this is to apply a Tikhonov regularization term to the matrix.

$$\begin{bmatrix} \frac{\partial^2 E_x}{\partial x^2} \Big|_{x_i} + \omega I & \gamma H^T \\ \gamma H & 0 + \omega I \end{bmatrix} \begin{bmatrix} \Delta x_i \\ \lambda \end{bmatrix} = \begin{bmatrix} -F_x(x_i) \\ \gamma H(x_{qk} - x_i) \end{bmatrix}. \quad (6.5)$$

where  $\omega$  is the regularization coefficient.

It will ensure the system remains positive definite at the cost of loosening the constraint at each step. This  $\ell^2$  norm penalty on displacements and Lagrange multipliers forces those vectors to take lower values than they would normally have. In the end, we get a more conservative step direction and a weaker enforcement of the constraint. The latter not been deranging as long as it doesn't influence the wrinkle periodicity. It however certainly give the skin a draping effect as opposed to a perfectly bonded stiffer layer.

Supported by the Tikhonov regularization, we gain access to simpler solvers such as conjugate gradient, and also we can improve on the overall conditioning by relying on an incomplete Cholesky preconditioner. Despite the usability of this formulation, we aim at one that will enforce the constraint in an exact fashion.

## 6.2 Surface Statics: A Revised Approach

We propose a yet different approach which consists of dividing the space between two orthogonal subspace, the one spanned by the constraint and the other by its null space. By limiting the search space of the linear solver to the subspace unconstrained by  $H$ , we withdraw from Lagrange multipliers. Moreover, the constraint is always trivially satisfied under such partitioning.

The subspaces spanned by the constraint is defined by the projector  $Q_1$  obtained from the QR-decomposition of the constraint matrix  $H$ .

$$H^T = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix} \quad (6.6)$$

where  $R \in R^{c \times c}$  is a upper triangular matrix,  $Q_1 \in R^{n \times c}$  the subspace projector,  $Q_2 \in R^{n \times (n-c)}$  the null space projector,  $0 \in R^{(n-c) \times c}$  and  $c$  the size of the coupling constraint  $H^{c \times n}$ .

Restating the linear system at heart of the  $i$ th linearization of the surface quasi-static step,

$$K_{x_i} \Delta x_i = -f_x(x_i)$$

such that (6.7)

$$H \Delta x_i = H(x_q - x_i)$$

where  $K_{x_i} = \left. \frac{\partial^2 E_x}{\partial x^2} \right|_{x_i}$  is the system energy Hessian evaluated at configuration estimate  $x_i$ . At the end of the step, we update the configuration estimate and obtain the configuration for the next linearization  $x_{i+1} = x_i + \Delta x_i$ . For the very first step of the static solve, the problem is linearized around  $x_0$  an artist provided configuration or simply the last solution of the last simulation frame.

Using this subspace, we can efficiently satisfy the constraint. Prior to solving for the static solution, we update the position of the constrained degrees of freedom

$$R^T Q_1^T \Delta x_0 = H(x_q - x_0) \tag{6.8}$$

The solution for the first update  $\Delta x_0$  is obtained directly with forward substitution

$$\begin{aligned} R^T \Delta y &= H(x_q - x_0) \\ Q_1 \Delta y &= \Delta x_0 \end{aligned} \tag{6.9}$$

The remaining refinements are done in  $Q_2$ , the constraint's null-space, and serve to solve the energy equation

$$Q_2 K_{x_i} \Delta x_i = -Q_2 f_x(x_i). \tag{6.10}$$

We solve the latter equation using a modified MINRES, an algorithm presented by Boxerman and Ascher [5, 6] originally on conjugate gradient.

We modified the solver to filter the solver search range by the orthogonal projector  $Q_2$  which only involves matrix-vector products. The use of preconditioning is still possible, but given the non-definitiveness of the system, we suggest the use of symmetrical successive overrelaxation preconditioning.

Conceptually the equation presented are well founded, However, it is unpractical to use the projectors  $Q_1$  and  $Q_2$ . Apart from the work required to form the projectors, the sparse property of  $H$  is not transmitted to  $Q$ . As illustrated in Figure 6–1, the orthogonal matrix  $Q$  is dense and contains more non-zero values than the system energy Hessian. Using it to perform multiplication is out of question.

Instead, we can use both  $R$  and  $H$  to replace  $Q$  in the previous equations. This substitution significantly improves the speed of the projection without weakening the stability of the algorithm.

We include to the precomputation the Q-less QR-decomposition of  $H$  to obtain  $R$ . Then both  $H$  and  $R$  matrices are ready to be used throughout the rest of the simulation. Instead of multiplying with the dense projector  $Q_1$ , we use this procedure

$$\begin{aligned}
 R^T \Delta y &= H(x_q - x_0) \\
 R \Delta y &= \Delta z \\
 H^T \Delta z &= \Delta x_0.
 \end{aligned}
 \tag{6.11}$$

We replace the projection by the backward substitution of  $R$  followed by the multiplication with the sparse  $H$ . Similarly, we can substitute the  $Q_2$  projection by  $(I - Q_1 Q_1^T)$  then to  $(I - H^T R^{-1} R^{-T} H)$ . The procedure for

$$Q_2 v = w
 \tag{6.12}$$

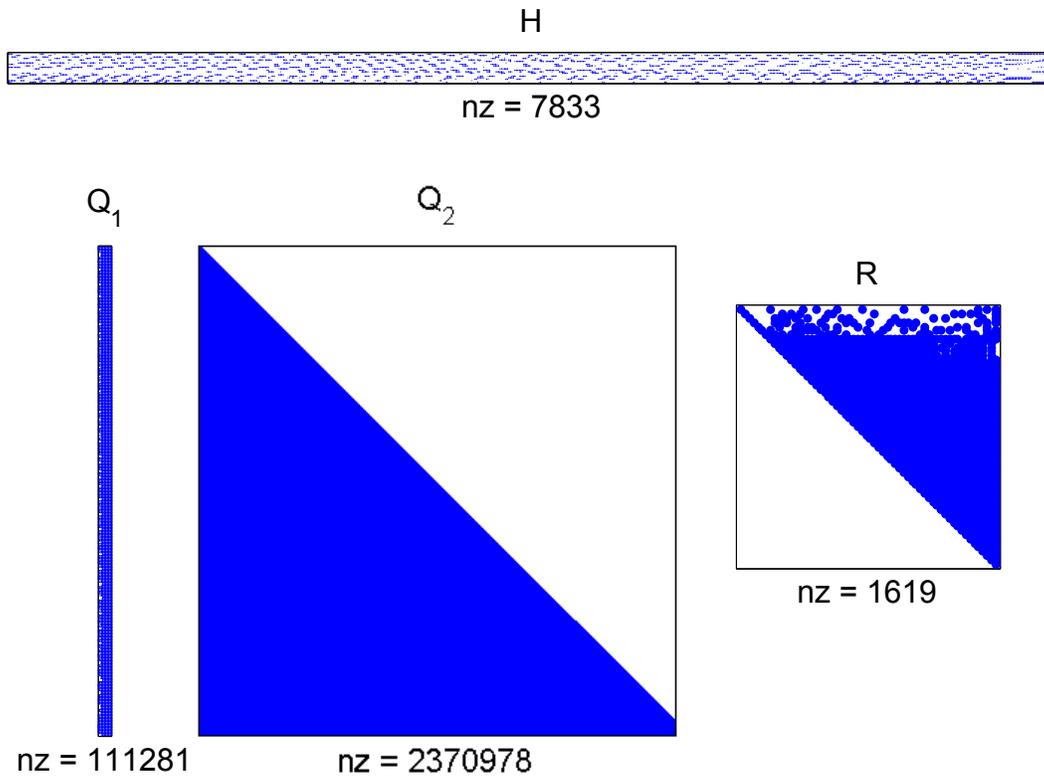


Figure 6–1: QR-decomposition of constraint  $H$  of the folded bar from Figures 5–1, 5–2 and 3–2 composed of  $c=62$  neighbourhoods and 6534 vertices. Every dot depicted represents a  $3 \times 3$  non-zero diagonal block matrix. The count non zero blocks is accumulated into  $nz$ . It shows that even for a relatively easy problem  $Q$  is dense and prohibitively expensive to use while  $H$  and  $R$  remain manageable.

is thus

$$\begin{aligned} R^T y &= H v \\ R z &= y \\ v - H^T z &= w. \end{aligned} \tag{6.13}$$

First, we multiply  $v$  with  $H$  and solve for  $y$  via forward substitution of  $R$  followed by a backward substitution to get  $z$  to be a multiplication and vector subtraction away from the projection  $w$  of the initial vector  $v$ . The procedure closely resembles the steps required to solve the constraint equation. For this reason, it is crucial to optimize the projection as much as possible by using the sparsity pattern of  $H$  and both forward, and backward substitutions to solve the matrix inversion of  $R$ .

### 6.3 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,

$$M \frac{\dot{q}_{k+1} - \dot{q}_k}{h} - F_q(q_{k+1}) - B^T F_x(x_{k+1}) = 0, \tag{6.14}$$

$$\frac{q_{k+1} - q_k}{h} - \dot{q}_{k+1} = 0. \tag{6.15}$$

We must linearize both force terms to perform Newton-Raphson iteration.

Interior forces at  $q_{k+1}$  are easily approximated by

$$F_q(q_{k+1}) \approx F_q(q_k) - \left. \frac{\partial^2 E_q}{\partial q^2} \right|_{q_k} (q_{k+1} - q_k), \tag{6.16}$$

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,

$$\begin{aligned}
F_x(x_{k+1}) &\approx F_x(x_k) + \left. \frac{\partial F_x}{\partial x} \right|_{x_k} (x_{k+1} - x_k), \\
&\approx F_x(x_k) - \left. \frac{\partial^2 E_x}{\partial x^2} \right|_{x_k} B(q_{k+1} - q_k). \tag{6.17}
\end{aligned}$$

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

$$\begin{aligned}
&\left[ M + h^2 K_q(q_k + h\dot{q}_{k+1}) + h^2 B^T K_x(x_k) B \right] \left[ \Delta \dot{q} \right] \\
&= h (F_q(q_k + h\dot{q}_{k+1}) + B^T (F_x(x_k) - h K_x(x_k) B \dot{q}_{k+1})) \tag{6.18}
\end{aligned}$$

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  $x_{q_{k+1}} = Bq_{k+1}$  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.

## CHAPTER 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.

#### 7.1 Timings

Table 7–1 shows information and timing for a variety of models and simulations that appears in figures throughout this work. The timings illustrate another advantage of the projection method for solving the system. It is faster from the first approach as long as the constraints are few. In general this should be a de facto situation as an over constrained system hinders the formation of wrinkles. Not only is this technique faster to iterate and converge, but it provides a strong guaranty that the constraint is fully satisfied at every step. The projection technique is relevant as long as the projection is numerically stable and the cost of computing a null-space projection is comparably small considering it occurs many thousands of times in a solve.

#### 7.2 Full Simulation Comparison

As part of validation, we compare the results of the embedded thin shell technique to a volumetric simulation of a slab discretized with a large number of volumetric elements. We go back to the well studied simple model for buckling exposed in chapter 4 as the foundation of the comparison. A thin slab of a flat skin attached on a soft foundation is compressed on its principal axis.

	$h$	$\mathbf{E}_x/\mathbf{E}_q$	$\lambda$	$c$	$n$	$n_x$	$n_q$	$t_{q\text{-setup}}$	$t_{H\text{-setup}}$	$t_x$	$t_{xA}$	$t_{Fx}$	$t_q$	$t_{qA}$	$t_{Fq}$	$t'_x$	$t'_{\text{proj}}$
pillow	0.0002	$10^2$	0.00404	2108	28194	19792	45	1584	1377	45690	521	14	45	5	4	68640	65
pillow	0.0004	$10^2$	0.00808	1488	12534	8352	45	982	1231	6328	303	6	37	4	3	1578	26
pillow	0.0005	$10^2$	0.01010	1046	3138	2088	45	361	329	639	47	1	35	4	2	164	13
pillow	0.0007	$10^4$	0.06566	98	28194	19792	45	1584	7902	36746	502	14	45	5	4	2257	0.6
pillow	0.0007	$10^4$	0.06566	100	12534	8352	45	982	1128	4536	219	5	37	4	3	955	0.3
pillow	0.0007	$10^4$	0.06566	95	3138	2088	45	361	234	576	40	1	35	4	2	467	0.2
pillow	0.0007	$10^4$	0.06566	87	636	420	45	163	89	69	7	1	34	3	2	58	0.1
bar	0.001	$10^2$	0.02021	33	6534	4352	8	280	805	2257	189	6	6	0.4	0.3	328	0.2
bar	0.0001	$10^2$	0.00202	82	6534	4352	8	296	649	2694	196	5	6	0.4	0.3	411	0.3
slab	0.001	$10^2$	0.02021	6	918	500	1		109	162	12	0.7	1	0.1	0.1	52	
slab	0.0002	$10^3$	0.00870	21	918	500	1		108	252	19	0.7	1	0.1	0.1	66	

Table 7–1: Example sizes and times for some of different simulations involving the 50 cm pillow in the supplementary video, the 80 cm bar from Figure 3–2, and the 50 cm slab from Figure 7–2. The skin thickness is  $h$  meters, the Young’s Modulus of skin and interior are  $\mathbf{E}_x$  and  $\mathbf{E}_q$  respectively,  $\lambda$  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 with Lagrange multipliers,  $t_{xA}$  is the assembly portion of the static solve time,  $t_{Fx}$  is the time to compute the shell forces. The timings of the projection’s technique are indicate under  $t'_x$ , and the actual time to project a vector into the null space of the constraint into  $t'_{\text{proj}}$ . 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 finite element lattice elastic forces. All times are in milliseconds.

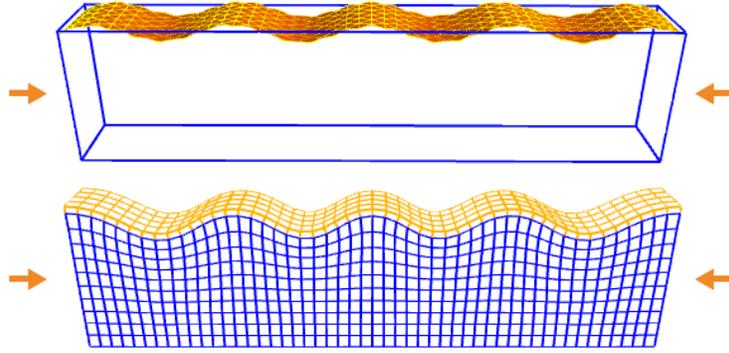


Figure 7–1: Wavelengths are measured in simulations of thin shells embedded into a single element (top), and in simulations of dense volumetric elements (bottom). The orange element represents the skin and possess different thickness and Young’s modulus than the interior’s blue elements.

Even though, the compression is sufficient for creating buckling in practice, computationally, buckling does not form under this simple setting. The system will instead reach a plausible static pose where the skin is compressed but remains flat and its stress is planar. This pose is ephemeral as a small out of plane perturbation brings the slab to buckle as expected. It is rarely observed in nature because the composition of the slab is not exactly the uniform, instead the material properties vary on the sample. We overcome this by applying a small load distributed around the centre of the slab during the beginning of the compression process. As the simulation progresses, we reduce the load until it is nonexistent by the time the slab is at a tenth of its final compression. An example simulation of the embedded thin shells and the dense volumetric simulation can be seen in Figure 7–1, while in Figure 7–2 we compare wrinkle wavelengths for a large variety of different skin thickness and Young’s modulus ratios. The different colours of the dots indicate different thickness, and we note that the models match well. Note that the size of the block and the number of elements limit the range of possible wrinkle wavelengths for this test. Wrinkles will not form when the wavelength approaches the width of the slab, and at high

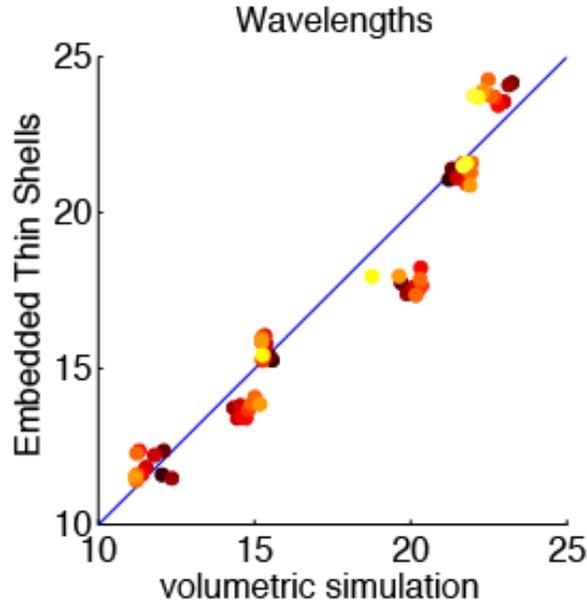


Figure 7–2: Wrinkle wavelengths compared across many material parameters for the embedded thin shells and the dense volume simulation.

frequencies the wrinkles take on a wavelength dictated by the finite element lattice resolution.

### 7.3 Critical Wavelength Comparison

Research on wrinkling often revolves around the modeling of the critical wavelength, such as the wavelength model of Equation 4.5 given thickness and elasticity parameters. We show in Figure 7–3 that the 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 the top left of Figure 7–2.

### 7.4 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

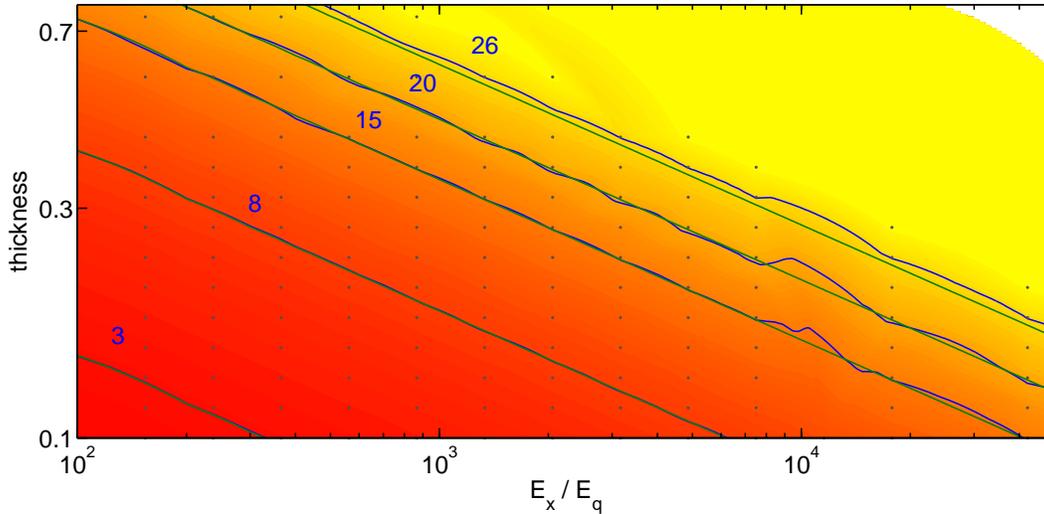


Figure 7–3: Wavelengths for different material parameters, as estimated from sample simulations (at grey dots). Wavelength isocontours (blue) closely match predictions of critical wavelength (green).

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 [23] 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 7–4 with a real example below. The figure also shows that the model produces the familiar flower patterns that are often seen in upholstered furniture.

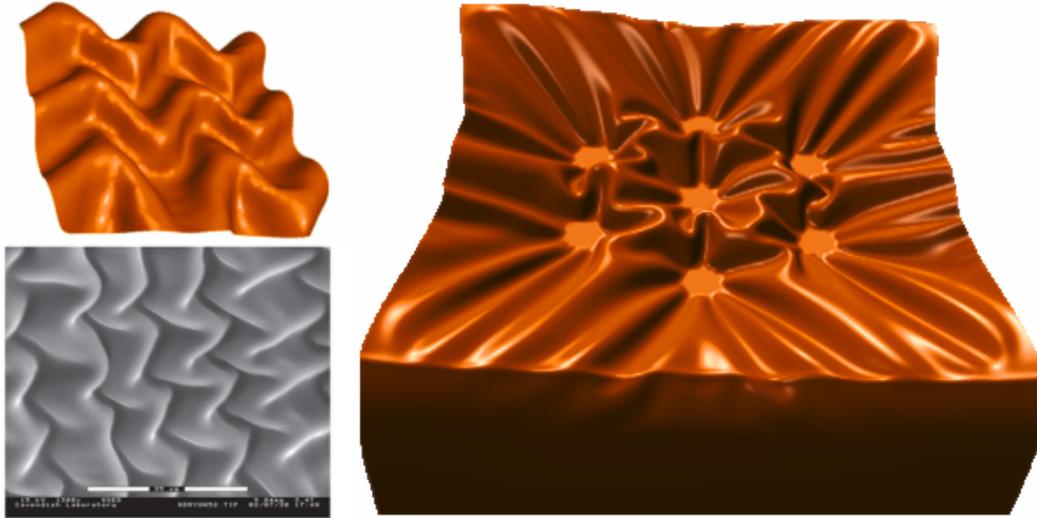


Figure 7–4: Left top, a Miura-ori pattern emerging after biaxial compression, and bottom, the same pattern in a thin film atop a thick elastic substrate that manifest in a drying slab of gelatin with a thin skin that forms naturally (from Mahadevan and Rica[2005] with permission). Right, distinctive flower patterns emerge with localized pressure, resembling deep buttoning on upholstered furniture.

## 7.5 Facial Wrinkles

One possible application for embedded thin shells is to produce wrinkles for character animation. Figure 7–5 shows wrinkles produced for different skin properties, resulting in different wavelengths.

The facial expressions are obtained by the use of muscle activation driven animation. First, a desired set of muscles are sculpted in their natural relaxed shape and position. The same muscles are then sculpted in a state of full activation. For the presented expression, we use the frowning action of the procerus and corrugator muscle responsible for the brow and nasal wrinkles coupled with the zygomatic raising the cheeks. Then the muscle relaxed shape is embedded in the lattice following the same process as for the surface mesh. We approximate the shape of the muscle at any activation level between the relaxed and the activated state by linearly blending the two shapes. We convert the muscle into forces by sampling the



Figure 7-5: 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 parameters (left) along with the large deformation embedded mesh (right).

muscle volume and exerting spring forces between the embedded sample and the target muscle activation level. 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.

## 7.6 Limitations

The lack of effective mass and the stiff nature of the skin of the embedded thin shells lead us to solve the skin statics. We deal with this non positive definite system by using modified MINRES. So far, the solver converges to solutions we judge acceptable. However, we have no proof as if the solver will always converge to a solution or at what rate it does so.

Yet another issue is that the static solutions of the skin of an animation has no guaranty to be temporally coherent. We promote coherence by choosing a safe time step size that favours a sequence of interior configurations that are spatially similar. In practice, a small time step is sufficient to preserve coherence of the skin from frame to frame as illustrated in the buttoned cushion and the squishy pillow from Figure 7–6 and Figure 7–7. Despite this, if a certain animation would cause problem, we suggest adding an extra coherence energy term to the skin static problem that would help close the gaps and force the solver to choose a solution closer to the last frame. We avoid doing this whenever possible, and only add the extra term sparingly. It can restrain the freedom of the skin and hinder the natural deformation of wrinkles.

Similar to most other high resolution simulation methods, embedded thin shell can suffer from slow solve times. We managed to find a good trade off, moving most of the simulation of volume’s finite elements to the surface to thin shells. The problem remains, we still must simulate the surface’s thousands of faces. Currently, we do offer a means for choosing how the surface be discretized. By learning the expected wrinkle wavelength from Equation 4.5, we can easily select a level of detail that cleanly permits formation of wrinkles. On the other hand, the task gets harder as soon as we complicate the model to contain varying material properties.

The constraint neighbourhoods selection algorithm is rudimentary and is only meant to show the validity of using frequency filters for coupling the thin shells with the coarse deformation lattice.

Inspired by the variational shape approximation developed by Cohen-Steiner et al. [10] which was adapted to be deformation aware in TRACKS [2], we can remesh surface to accommodate the critical wrinkle

wavelength based on varying material properties and object geometry. In addition, the selection of constraint clusters could be chosen accordingly. A modified variational shape approximation is a method for reducing the number of faces required to model the surface, without having a negative effect on the wrinkle formation.

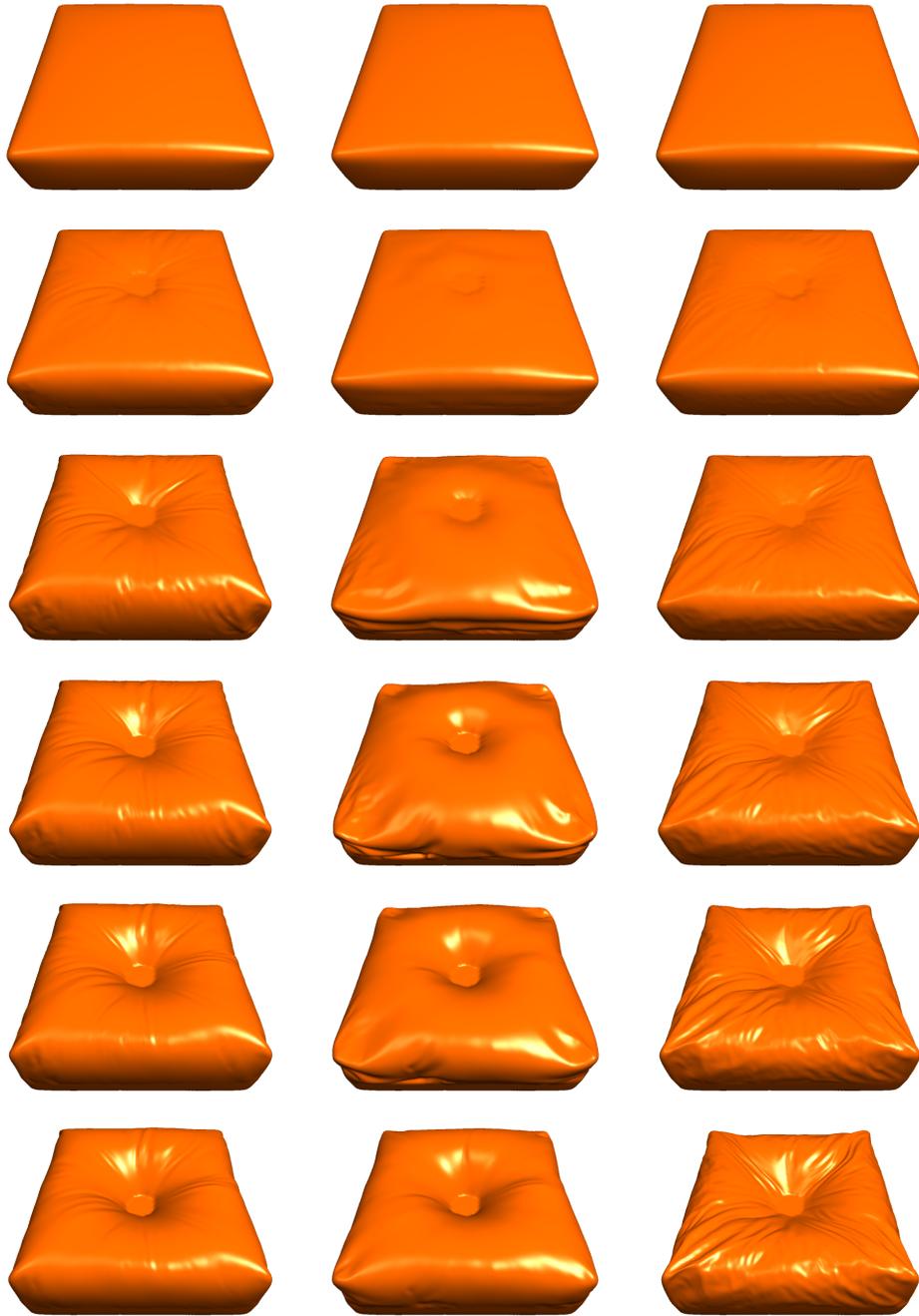


Figure 7-6: We compare the effect of different set of properties from the linen cloth ( $E = 25\text{GPa}$ ,  $h = 1\text{mm}$ ) on a foam block (sequence on the left). The centre sequence has a supple but very thick skin ( $E = 50\text{MPa}$ ,  $h = 8\text{mm}$ ) while the rightmost sequence has a thin and rigid skin ( $E = 70\text{GPa}$ ,  $h = 50\mu\text{m}$ ).

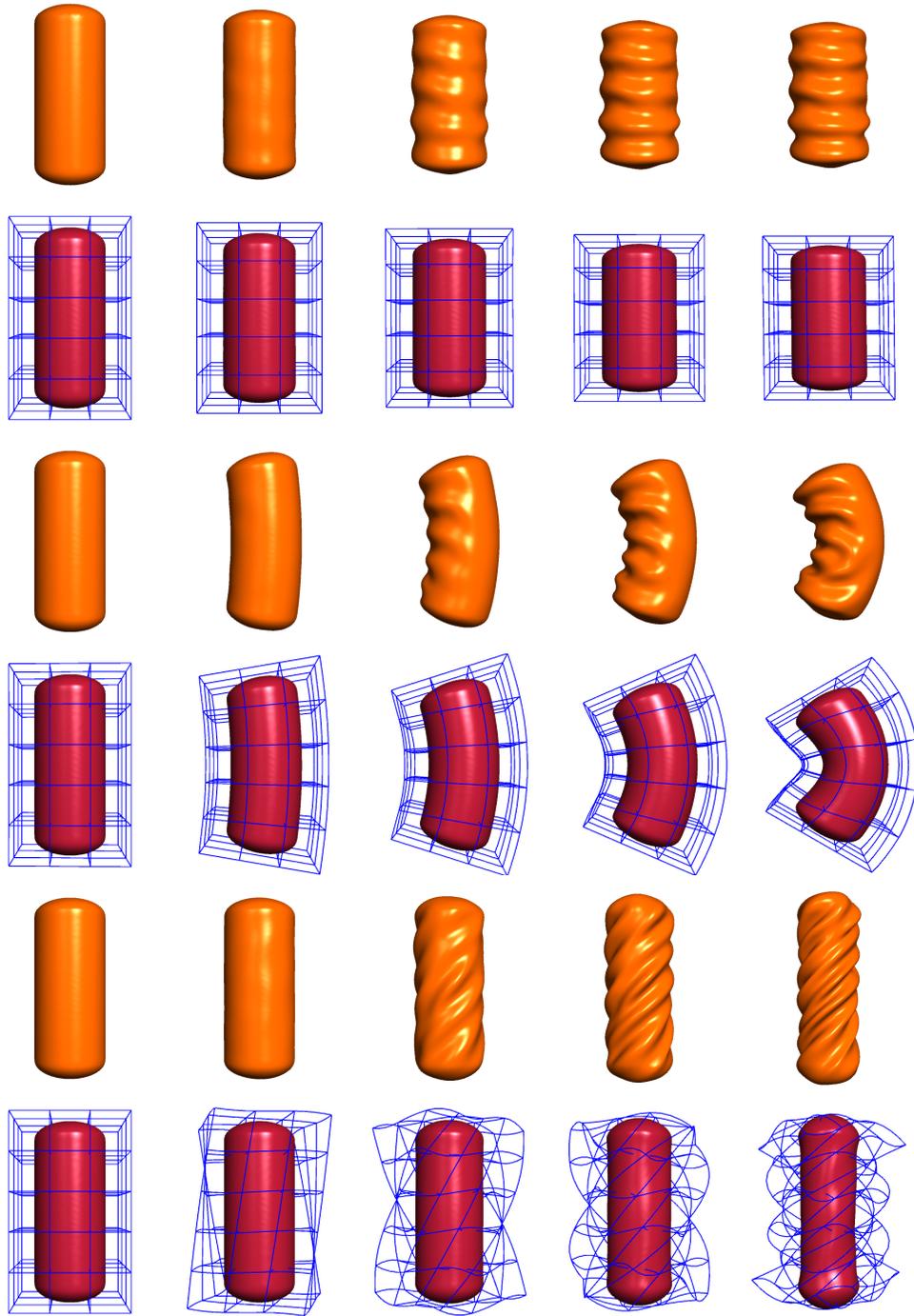


Figure 7–7: We expose the effect of different deformation to wrinkling. The sequences of the shell (orange) are compared to the embedded mesh (red) and the interior’s finite elements (blue).

## CHAPTER 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 couple the high resolution shell and 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 the static solver, such as adding a preconditioner to the modified MINRES or to substitute MINRES for a more specific solver optimized for linear constraints. The constraint matrix construction and cluster selection are likely to profit from the variational shape approximation. Furthermore, our largest examples will also benefit from parallelization techniques. Multi-resolution techniques hold considerable promise for reducing the computation time. Adaptive simulation, and more recently Narain et al. [29] with their work on anisotropic remeshing of clothes, would nicely augment embedded thin shells. The rendering of the highly detailed surface must not be neglected either. Embedded thin shells are a strong candidate for precomputed radiance transfer for its intrinsic coarse-to-fine structure. In sum, we hope that embedded thin

shells can become a fast alternative to both simulate and render complex wrinkling in character faces and objects.

## References

- [1] D. Baraff and A. Witkin. Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 43–54. ACM, 1998.
- [2] M. Bergou, S. Mathur, M. Wardetzky, and E. Grinspun. Tracks. In *ACM Transactions on Graphics (TOG)*, volume 26, page 50. ACM, 2007.
- [3] M. Bergou, M. Wardetzky, D. Harmon, D. Zorin, and E. Grinspun. A quadratic bending model for inextensible surfaces. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, pages 227–230, 2006.
- [4] J.F. Blinn. Simulation of wrinkled surfaces. *SIGGRAPH Comput. Graph.*, 12(3):286–292, August 1978.
- [5] E. Boxerman. Speeding up cloth simulation. Master's thesis, The University of British Columbia, 2003.
- [6] E. Boxerman and U. Ascher. Decomposing cloth. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 153–161. Eurographics Association, 2004.
- [7] R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '02, pages 594–603, 2002.
- [8] R. Bridson, S. Marino, and R. Fedkiw. Simulation of clothing with folds and wrinkles. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '03, pages 28–36, 2003.
- [9] S. Capell, S. Green, B. Curless, T. Duchamp, and Z. Popović. Interactive skeleton-driven dynamic deformations. *ACM Trans. Graph.*, 21(3):586–593, 2002.
- [10] D. Cohen-Steiner, P. Alliez, and M. Desbrun. Variational shape approximation. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 905–914. ACM, 2004.

- [11] D.A. Danielson. Human skin as an elastic membrane. *Journal of Biomechanics*, 6(5):539–546, 1973.
- [12] P. Faloutsos, M. van de Panne, and D. Terzopoulos. Dynamic free-form deformations for animation synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 3(3):201–214, 1997.
- [13] J. Genzer and J. Groenewold. Soft matter with hard skin: From skin wrinkles to templating and material characterization. *Soft Matter*, 2(4):310–323, 2006.
- [14] E. Grinspun, A.N. Hirani, M. Desbrun, and P. Schröder. Discrete shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 62–67, 2003.
- [15] G. Irving, J. Teran, and R. Fedkiw. Invertible finite elements for robust simulation of large deformation. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 131–140. Eurographics Association, 2004.
- [16] G. Irving, J. Teran, and R. Fedkiw. Tetrahedral and hexahedral invertible finite elements. *Graphical models*, 68(2):66–89, 2006.
- [17] J. Jimenez, J.I. Echevarria, C. Oat, and D. Gutierrez. *GPU Pro 2*, chapter Practical and Realistic Facial Wrinkles Animation, pages 15–27. AK Peters Ltd., 2011.
- [18] L. Kavan, D. Gerszewski, A.W. Bargteil, and P.P. Sloan. Physics-inspired upsampling for cloth simulation in games. In *ACM Transactions on Graphics (TOG)*, volume 30, page 93, 2011.
- [19] C.T. Kelley. *Solving Nonlinear Equations with Newton’s Method*. Fundamentals of Algorithms. Society for Industrial and Applied Mathematics, 1987.
- [20] C. Larboulette and M.P. Cani. Real-time dynamic wrinkles. In *Computer Graphics International 2004, CGI 2004, June, 2004*, pages 522–525. IEEE, June 2004.
- [21] B. Lévy and Y. Liu. Lp centroidal Voronoi tessellation and its applications. *ACM Trans. Graph.*, 29(4):119:1–119:11, 2010.
- [22] N. Magnenat-Thalmann, P. Kalra, J. Luc Leveque, R. Bazin, D. Batisse, and B. Querleux. A computational skin model: fold and wrinkle formation. *Information Technology in Biomedicine, IEEE Transactions on*, 6(4):317–323, 2002.
- [23] L. Mahadevan and S. Rica. Self-organized origami. *Science*, 307(5716):1740–1740, 2005.

- [24] A. McAdams, Y. Zhu, A. Selle, M. Empey, R. Tamstorf, J. Teran, and E. Sifakis. Efficient elasticity for character skinning with contact and collisions. *ACM Trans. Graph.*, 30(4):37:1–37:12, 2011.
- [25] J. Mezger, B. Thomaszewski, S. Pabst, and W. Straßer. Interactive physically-based shape editing. *Computer Aided Geometric Design*, 26(6):680–694, 2009.
- [26] N. Molino, Z. Bao, and R. Fedkiw. A virtual node algorithm for changing mesh topology during simulation. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 385–392, 2004.
- [27] M. Müller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler. Stable real-time deformations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '02, pages 49–54, 2002.
- [28] M. Müller and M. Gross. Interactive virtual materials. In *Proceedings of Graphics Interface 2004*, GI '04, pages 239–246, 2004.
- [29] R. Narain, A. Samii, and J.F. O'Brien. Adaptive anisotropic remeshing for cloth simulation. *ACM Transactions on Graphics (TOG)*, 31(6):152, 2012.
- [30] M. Nesme, P.G. Kry, L. Jeřábková, and F. Faure. Preserving topology and elasticity for embedded deformable models. In *ACM Transactions on Graphics (TOG)*, volume 28, page 52. ACM, 2009.
- [31] M. Nesme, P.G. Kry, L. Jeřábková, and F. Faure. Preserving topology and elasticity for embedded deformable models. *ACM Trans. Graph.*, 28(3):52:1–52:9, 2009.
- [32] C.C. Paige and M.A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, 12(4):617–629, 1975.
- [33] E.G. Parker and J.F. O'Brien. Real-time deformation and fracture in a game environment. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '09, pages 165–175, 2009.
- [34] T. Patterson, N. Mitchell, and E. Sifakis. Simulation of complex nonlinear elastic bodies using lattice deformer. *ACM Transactions on Graphics (TOG)*, 31(6):197, 2012.
- [35] D. Rohmer, T. Popa, M.P. Cani, S. Hahmann, and A. Sheffer. Animation wrinkling: augmenting coarse cloth simulations with

- realistic-looking wrinkles. *ACM Trans. Graph. (SIGGRAPH ASIA)*, 29(6):157:1–157:8, December 2010.
- [36] T.W. Sederberg and S.R. Parry. Free-form deformation of solid geometric models. *SIGGRAPH Comput. Graph.*, 20(4):151–160, August 1986.
- [37] M. Seiler, J. Spillmann, and M. Harders. Enriching coarse interactive elastic objects with high-resolution data-driven deformations. In *Proceedings of the 11th ACM SIGGRAPH / Eurographics conference on Computer Animation*, pages 9–17, 2012.
- [38] E. Sifakis, I. Neverov, and R. Fedkiw. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Trans. Graph.*, 24(3):417–425, July 2005.
- [39] E. Sifakis, T. Shinar, G. Irving, and R. Fedkiw. Hybrid simulation of deformable solids. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 81–90, 2007.
- [40] J. Teran, E. Sifakis, G. Irving, and R. Fedkiw. Robust quasistatic finite elements and flesh simulation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 181–190. ACM, 2005.
- [41] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *SIGGRAPH Comput. Graph.*, 21(4):205–214, August 1987.
- [42] D. Terzopoulos and K. Waters. Physically-based facial modelling, analysis, and animation. *The journal of visualization and computer animation*, 1(2):73–80, 1990.
- [43] D. Terzopoulos and A. Witkin. Physically based models with rigid and deformable components. *IEEE Comput. Graph. Appl.*, 8(6):41–51, November 1988.
- [44] B. Thomaszewski, M. Wacker, and W. Straßer. A consistent bending model for cloth simulation with corotational subdivision finite elements. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation, SCA '06*, pages 107–116, 2006.
- [45] S.P. Timoshenko and J.M. Gere. *Theory of Elastic Stability*. Dover Civil and Mechanical Engineering Series. Dover Publications, 2009.

- [46] K. Venkataraman, S. Lodha, and R. Raghavan. A kinematic-variational model for animating skin with wrinkles. *Computers and Graphics*, 29(5):756 – 770, 2005.
- [47] Yu Wang, Charlie CL Wang, and Matthew MF Yuen. Fast energy-based surface wrinkle modeling. *Computers & Graphics*, 30(1):111–125, 2006.
- [48] C. Wojtan and G. Turk. Fast viscoelastic behavior with thin features. In *ACM SIGGRAPH 2008 papers*, SIGGRAPH '08, pages 47:1–47:8, 2008.