ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2022
D. L. Michels and S. Pirk
(Guest Editors)

Volume 41 (2022), Number 8

A Second Order Cone Programming Approach for Simulating Biphasic Materials

Pengbin Tang[1], Stelian Coros[2] and Bernhard Thomaszewski[2]

[1]Université de Montréal, Canada
[2]ETH Zürich, Switzerland

Abstract

Strain limiting is a widely used approach for simulating biphasic materials such as woven textiles and biological tissue that exhibit a soft elastic regime followed by a hard deformation limit. However, existing methods are either based on slowly converging local iterations, or offer no guarantees on convergence. In this work, we propose a new approach to strain limiting based on second order cone programming (SOCP). Our work is based on the key insight that upper bounds on per-triangle deformations lead to convex quadratic inequality constraints. Though nonlinear, these constraints can be reformulated as inclusion conditions on convex sets, leading to a second order cone programming problem—a convex optimization problem that a) is guaranteed to have a unique solution and b) allows us to leverage efficient conic programming solvers. We first cast strain limiting with anisotropic bounds on stretching as a quadratically constrained quadratic program (QCQP), then show how this QCQP can be mapped to a second order cone programming problem. We further propose a constraint reflection scheme and empirically show that it exhibits superior energy-preservation properties compared to conventional end-of-step projection methods. Finally, we demonstrate our prototype implementation on a set of examples and illustrate how different deformation limits can be used to model a wide range of material behaviors.

1. Introduction

From biological tissues in plants and animals to woven textiles and 3D-printed chainmail—many natural and human-made materials exhibit complex stress-strain responses characterized by strong nonlinearities and direction-dependent behavior. For example, due to their yarn-level structure, many textiles initially exhibit only weak resistance to deformation but stiffen rapidly beyond a direction-dependent threshold. Another extreme example is given by the 3D-printed chainmail shown in Fig. 1. One way of modeling this *biphasic* behavior is to use concepts from nonlinear elasticity in combination with experimentally acquired data. An alternative approach that enjoys widespread use in computer animation is to use *strain limiting*, which combines a soft elastic material with hard constraints on the maximum allowed deformation.

Many strain limiting methods enforce deformation limits per spring or element, using, e.g., Gauss-Seidel like iterations [P*95, BFA02, TPS09]. While simplicity is an advantage of this local approach, convergence is often exceedingly slow, in particular if tight deformation bounds are used. Another class of methods combines individual constraints into a globally-coupled problem. While this strategy can greatly accelerate convergence, existing methods either solve only a linearized version of the problem [GHF*07, PCH*13] or resort to nonlinear programming techniques to deal with potentially nonconvex constraints [NSO12, JLGF17]. In either case, there are no formal guarantees on convergence.

In this work, we propose a new approach to strain limiting based on second order cone programming (SOCP). Using a finite element discretization, we model bounds on triangle deformations in given material directions using quadratic inequality constraints. Our work is based on the key insight that bounds on compression lead to nonconvex constraints, whereas bounds on stretching lead to convex constraints. Though nonlinear, stretching constraints can be reformulated as inclusion conditions on convex sets, leading to a second order cone programming problem—a convex optimization problem that a) is guaranteed to have a unique solution and b) can be solved much more efficiently than a general nonlinear program [ART03].

To develop our formulation, we first cast strain limiting with anisotropic bounds on stretching as a quadratically constrained quadratic program (QCQP), then show how this QCQP can be mapped to a second order cone programming problem. We further propose a constraint reflection scheme and empirically show that it exhibits superior energy-preservation properties compared to conventional end-of-step projection methods. Finally, we demonstrate our prototype implementation on a set of examples and illustrate how different anisotropic deformation limits can be used to model a wide range of material behaviors.

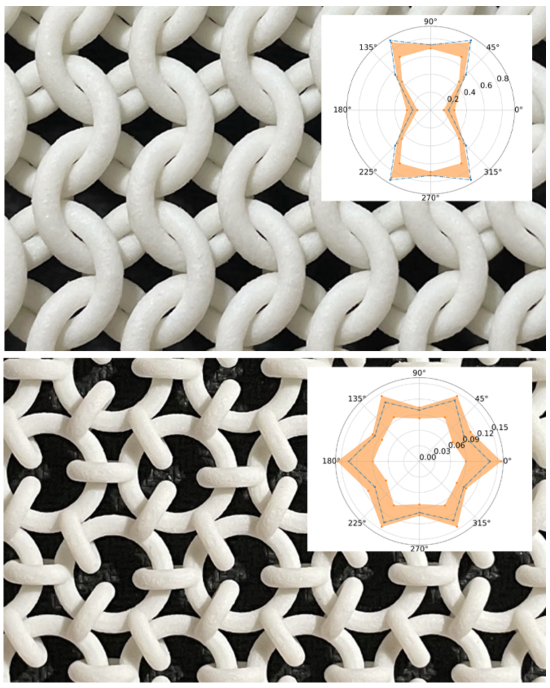

Figure 1: *Two examples of 3D-printed chainmail with experimentally measured deformation limits. Orange areas in the plots indicate the variation of experimental data.*

2. Related Work

Modeling the mechanical behavior of materials is a problem that received much attention in computer animation; see, e.g., [XSZB15, LB15] for recent examples. For textiles, the spectrum ranges from simple mass spring systems with manually specified stiffness coefficients [P*95, BFA02] to continuum mechanics models with material parameters extracted from real-world measurements [WOR11, MBT*12]. Instead of modeling fabrics using general nonlinear elasticity, another approach is to approximate them as biphasic materials that exhibit low stiffness for small deformations followed by a quasi-inextensible limit. Strain limiting is a way of modeling these biphasic materials.

The idea of strain limiting goes back to the work of Provot [P*95] who experimented with mass-spring simulations using explicit time integration. Using lower stiffness coefficients allowed for larger time steps, but led to undesirable over-elongation effects. Provot corrected for these artifacts by iteratively adjusting vertex positions after every integration step. This relatively simple idea has been widely successful and was used and extended in numerous works. For example, Bridson et al. [BFA02] integrate strain limiting and collision handling as layers in a velocity filtering stack. Thomaszewski et al. [TPS09] extended strain limiting to the continuum-mechanics setting by enforcing bounds on the entries of the co-rotated Cauchy strain tensor. Subsequent work introduced limits on the principal strain to achieve isotropic [WOR10] and anisotropic [HCPO13] behaviors, and proposed extensions to quadratic strain measures [MCKM15].

While all the above approaches enforce strain limits locally by iterating over individual springs and elements, another line of work aimed at globally-coupled enforcement based on constrained optimization techniques. For example, Goldenthal et al. [GHF*07] propose a fast projection scheme in which constraints are linearized and combined into a global system. The corresponding system is solved repeatedly until the constraint violation is sufficiently small. While this approach is computationally more efficient than simpler per-element iterations, there are no formal convergence guarantees. Nevertheless, fast projection has been extended to triangle meshes [EB08], and it has been used for augmenting conventional simulators with energy-momentum conservation [DLL*18].

In the context of haptic simulation, Perez et al. [PCH*13] propose a globally-coupled treatment of strain limiting and frictional contact, combining linearized constraints into a linear complementarity problem that is solved with Projected Gauss-Seidel. The resulting method is fast enough to enable haptic interaction, although tight constraint enforcement cannot be guaranteed due to the linearized formulation.

In contrast, Jin et al. [JLGF17] propose a fully nonlinear formulation in which inequality constraints on edge lengths are enforced in an iterative fashion using an active set-like algorithm. While Jin et al. stress the importance of their constraint relaxation and tightening scheme to avoid undesirable local minima, our formulation is convex and is therefore guaranteed to converge to a single global optimum.

Generalizing the idea of strain limiting, Position-based Dynamics (PBD) is a constraint-based animation technique that has been widely successful in computer animation; see [BMM15] for an overview. In analogy to developments in strain limiting, Bouaziz et al. [BML*14] introduced a method that combines the local constraint projections of PBD into a global solve. Overby et al. [OBLN17] showed that the formulation by Bouaziz et al. is equivalent to solving a corresponding constraint optimization problem using the Alternating Direction of Multipliers Method (ADMM) and can thus benefit from progress in ADMM solvers [ZPOD19]. While ADMM is often the method of choice when it comes to solving non-convex optimization problems, our formulation is convex from the ground up. Nevertheless, some SOCP solvers are implemented using ADMM on their inside.

3. Theory

We consider discrete elastic surfaces represented by triangle meshes with two sets of vertices, $\bar{\mathbf{x}}$ and $\mathbf{x}$, holding undeformed and deformed positions, respectively. Let $\mathbf{x}^i_k$, $\bar{\mathbf{x}}^k_i$ denote deformed and undeformed vertex positions for a given triangle $\mathcal{T}_i$. Using a finite element discretization over the space of piece-wise linear functions, the deformation gradient $\mathbf{F} \in \mathbb{R}^{3\times 2}$ is the unique matrix that maps undeformed edge vectors to their deformed counterparts as

$$(\mathbf{x}^i_k - \mathbf{x}^i_l) = \mathbf{F}(\bar{\mathbf{x}}^i_k - \bar{\mathbf{x}}^i_l) \ . \tag{1}$$

It is evident from this expression that $\mathbf{F}$ is a linear function of the deformed positions $\mathbf{x}$—a fact that allows us to formulate bounds on stretching as quadratic inequality constraints.

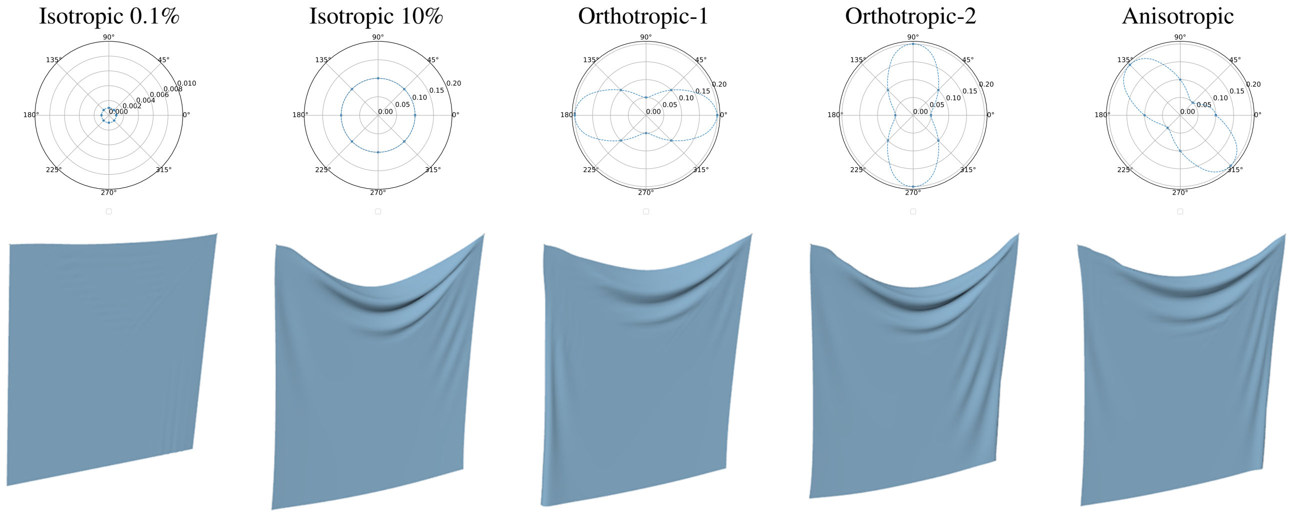

Figure 2: *Our method can enforce stretching limits in arbitrary material directions including isotropic, orthotropic, and fully anisotropic constraints. Top row: polar plots showing input curves for deformation limits and sample locations for which constraints are enforced. Bottom row: simulation results for a static drape test using a mesh resolution of* 60×60.

3.1. Deformation Limits as Quadratic Constraints

Orthotropic and anisotropic materials show variations in their stress-strain response as a function of the direction of deformation. Likewise, deformation limits are generally direction-dependent. In order to properly enforce these limits, we must first establish how to measure deformation in arbitrary directions. Using the deformation gradient from (1) for a given deformed triangle, we can quantify the directional deformation for any unit-length material-space direction $\bar{\mathbf{d}} \in \mathbb{R}^2$ as the squared length of the transformed vector, i.e.,

$$s(\bar{\mathbf{d}}) = \mathbf{d}^T \mathbf{d} = \bar{\mathbf{d}}^T \mathbf{F}^T \mathbf{F} \bar{\mathbf{d}} \ . \tag{2}$$

Values $s(\bar{\mathbf{d}}) \geq 1$ indicate stretching, whereas $s(\bar{\mathbf{d}}) \leq 1$ implies compression. By parameterizing the direction $\bar{\mathbf{d}}$ through a single scalar, $\bar{\mathbf{d}}$=$\bar{\mathbf{d}}(\alpha_i)$, we obtain a continuous measure of deformation in polar space. We note that, for any direction α, this measure is a quadratic function of the element's deformed positions $\mathbf{x}_k^i$. Next, we discuss how to impose limits on deformations.

We assume that deformation limits are provided through m samples, (α_i, b_i), consisting of an angle $\alpha_i \in [-\pi/2, \pi/2]$ and a stretch limit b_i. We then impose a set of quadratic inequality constraints, i.e., $s(\alpha_i, \mathbf{x}) \leq b^2(\alpha_i)\ \forall i$. Before we can proceed to a conic reformulation, we must first transform these constraints into a different form. To this end, we observe that for any material-space direction $\bar{\mathbf{d}}_i$, the corresponding deformed vector $\mathbf{d}_i(\mathbf{x})$ is a linear function of $\mathbf{x}$ and we write

$$\mathbf{d}_i(\mathbf{x}) = \mathbf{F}(\mathbf{x})\bar{\mathbf{d}}_i = \mathbf{A}_i \mathbf{x} \ , \tag{3}$$

where $\mathbf{A}_i = \nabla_{\mathbf{x}}(\mathbf{F}\bar{\mathbf{d}}_i) \in \mathbb{R}^{3\times 9}$ holds products between the partial derivatives of $\mathbf{F}$ and components of $\bar{\mathbf{d}}_i$ (see Appendix). We can now rewrite the quadratic constraints as

$$s_i(\mathbf{x}) = s(\alpha_i, \mathbf{x}) = ||\mathbf{A}_i \mathbf{x}||^2 = \mathbf{x}^T \mathbf{A}_i^T \mathbf{A}_i \mathbf{x} \leq b_i^2 \ . \tag{4}$$

3.2. Constraint Projection as QCQP

We simulate biphasic materials using a step-and-project approach [GHF*07]: we first compute candidate positions $\hat{\mathbf{x}}$ using a standard implicit time stepping scheme without deformation constraints. We then project these candidate positions onto the constraint manifold by computing the closest configuration that satisfies all constraints, i.e.,

$$\min \frac{1}{2}||\mathbf{x} - \hat{\mathbf{x}}||^2 \quad \text{such that} \qquad (5)$$
$$\mathbf{x}^T \mathbf{A}_i^T \mathbf{A}_i \mathbf{x} \leq b_i^2\ \forall i \ .$$

The above expression describes a quadratically-constrained quadratic program (QCQP). Since the matrices $\mathbf{A}^T \mathbf{A}$ are symmetric positive semi-definite, this QCQP is convex and thus has a single global optimum. Conversely, imposing lower bounds on deformation corresponds to multiplying the quadratic constraint by -1. The matrix $-\mathbf{A}_i^T \mathbf{A}_i$ is evidently negative semi-definite and the constraint is no longer convex. Interestingly, the bilateral version of this constraint is likewise nonconvex. This can be understood geometrically since the feasible set for the quadratic equality constraint corresponds to an ellipsoidal surface which is not a convex set: for any two points on the surface, the line connecting the two points does not belong to the feasible set. These observations are valuable as they assert that our projection problem—which enforces bounds on stretching but not compression—has a unique solution. This is a crucial property that previous formulations did not possess or expose. However, in order to leverage the full computational advantages that this setting offers, we must transform (5) into a conic formulation as follows.

3.3. Conic Reformulation

Conic Programs (CP) are optimization problems in which inequality constraints are expressed as set inclusion conditions on convex cones [LVBL98, AG03]. A set $\mathcal{K} \subset \mathbb{R}^n$ is called a convex cone if for each $\mathbf{y}, \mathbf{z} \in \mathcal{K}$ we have $\mathbf{y}+\mathbf{z} \in \mathcal{K}$ and for each $\mathbf{y} \in \mathcal{K}$ we have $\alpha\mathbf{y} \in \mathcal{K}$ for all $\alpha \leq 0$. We focus on quadratic (i.e., second-order) cones, which are defined as

$$\mathcal{Q}^{n+1} = \{(t, \mathbf{y}) \in \mathbb{R}^{n+1} \mid ||\mathbf{y}||_2 \leq t\} \, . \tag{6}$$

While quadratic cones directly allow for Euclidean norm constraints, it is often more convenient to impose constraints on squared norms. To this end, we use so-called rotated second-order cones, which are defined as

$$\mathcal{Q}_r^{n+1} = \{(u, v, \mathbf{y}) \in \mathbb{R}^{n+2},\ u, v \geq 0 \mid ||\mathbf{y}||_2^2 \leq 2uv\} \, . \tag{7}$$

To convert the quadratic inequality constraints (4) into equivalent second-order cone constraints, we first introduce two auxiliary variables t_i and $\mathbf{y}_i$ such that

$$\begin{aligned} t_i &= b_i^2 \, , \\ \mathbf{y}_i &= \mathbf{A}_i \mathbf{x} \, . \end{aligned} \tag{8}$$

The quadratic constraint is now readily expressed in a second-order cone form as

$$\mathbf{x}^T \mathbf{A}_i^T \mathbf{A}_i \mathbf{x} \leq b_i^2 \iff (\tfrac{1}{2}, t_i, \mathbf{y}_i) \in \mathcal{Q}_r^{k+2} \, . \tag{9}$$

With all constraints reduced to conic form, we must now transform the quadratic objective function in (5) into a linear form. This can be accomplished by first writing out the quadratic objective as

$$\min \frac{1}{2}\mathbf{x}^T\mathbf{x} - \mathbf{x}^T\hat{\mathbf{x}} + \frac{1}{2}\hat{\mathbf{x}}^T\hat{\mathbf{x}} \, . \tag{10}$$

Omitting the constant term and introducing an additional auxiliary variable t_0, this quadratic objective is equivalent to

$$\min_{\mathbf{x}, t_0} \quad t_0 - \hat{\mathbf{x}}^T\mathbf{x} \quad \text{s.t.} \quad \mathbf{x}^T\mathbf{x} \leq 2t_0 \, , \tag{11}$$

where the quadratic inequality constraint can be directly translated into conic form as outlined above. Combining the conic reformulations for the objective and all constraints, we arrive at

$$\begin{aligned} \min_{\mathbf{x}, t_0, t_i, \mathbf{y}_i} \quad & t_0 - \hat{\mathbf{x}}^T\mathbf{x} \quad \text{such that} \\ & (1, t_0, \mathbf{x}) \in \mathcal{Q}_r^{n+2} \\ & t_i = b_i^2 \quad \forall i, \\ & \mathbf{y}_i = \mathbf{A}_i\mathbf{x} \quad \forall i, \\ & (\tfrac{1}{2}, t_i, \mathbf{y}_i) \in \mathcal{Q}_r^{k+2} \quad \forall i \, . \end{aligned} \tag{12}$$

With the above formulation, we have converted our constraint projection problem into SOCP form. It is therefore guaranteed to have a unique solution, and that solution can be computed with any SOCP solver (we use Mosek 9.3).

4. Time Stepping

When implemented as a post-step velocity filter in the spirit of [BFA02], our SOCP formulation integrates seamlessly with standard time-stepping schemes and collision response methods. This step-and-project (SAP) approach generally leads to stable and visually pleasing motion. We notice, however, that it largely inherits the numerical dissipation of lower-order implicit integration schemes, which manifests as a loss of kinetic energy and dynamic detail over time; see the accompanying video.

Algorithm 1 Step-and-Reflect

```
1: procedure StepAndReflect($\mathbf{x}^0, \mathbf{v}^0$)
2:     $\tilde{\mathbf{x}}^{1/2}, \tilde{\mathbf{v}}^{1/2} \leftarrow$ ImplicitEulerStep($\mathbf{x}^0, \mathbf{v}^0, \frac{\Delta t}{2}$)
3:     $\mathbf{x}^{1/2} \leftarrow$ StrainLimitingSOCP($\tilde{\mathbf{x}}^{1/2}$)
4:     $\mathbf{v}^{1/2} = \frac{\mathbf{x}^{1/2} - \mathbf{x}^0}{\Delta t/2}$
5:     $\hat{\mathbf{x}}^{1/2} = 2\mathbf{x}^{1/2} - \tilde{\mathbf{x}}^{1/2}, \hat{\mathbf{v}}^{1/2} = \mathbf{v}^{1/2}$
6:     $\tilde{\mathbf{x}}^1, \tilde{\mathbf{v}}^1 \leftarrow$ ImplicitEulerStep($\hat{\mathbf{x}}^{1/2}, \hat{\mathbf{v}}^{1/2}$)
7:     $\mathbf{x}^1 \leftarrow$ StrainLimitingSOCP($\tilde{\mathbf{x}}^1$)
8:     $\mathbf{v}^1 = \frac{\mathbf{x}^1 - \hat{\mathbf{x}}^{1/2}}{\Delta t/2}$
```

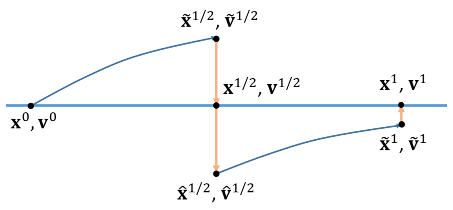
$\tilde{\mathbf{x}}^{1/2}, \tilde{\mathbf{v}}^{1/2}$
$\mathbf{x}^{1/2}, \mathbf{v}^{1/2}$
$\mathbf{x}^1, \mathbf{v}^1$
$\mathbf{x}^0, \mathbf{v}^0$
$\tilde{\mathbf{x}}^1, \tilde{\mathbf{v}}^1$
$\hat{\mathbf{x}}^{1/2}, \hat{\mathbf{v}}^{1/2}$

Figure 3: *A schematic visualization of our quasi-symmetric step-and-reflect method. We first step to the middle of the time interval, where we apply twice the correction necessary to satisfy deformation constraints. We then integrate from mid-step to the end of the interval and apply a standard projection step.*

To improve upon this behavior, we follow the idea by Zehnder et al. [ZNT18] and experiment with a quasi-symmetric constraint reflection scheme that we refer to as *step-and-reflect (SAR)*.

To this end, we first integrate to the middle of the time interval, where we enforce constraints by solving (12). We then apply twice the correction required to satisfy all constraints which, visually, reflects the current configuration to the other side of the constraint manifold (i.e., the boundary between feasible and infeasible configurations). We step from this reflected configuration to the end of the interval, where we apply a standard constraint projection. See Fig. 3 for a visual summary and Algorithm 1 for a pseudo-code description of this scheme.

The rationale of this step-and-reflect approach is that, in analogy to symmetric (i.e., time-reversible) integration schemes, the quasi-symmetric arrangement of constraint correction and integration steps should lead to better energy conservation [HLW06]. Similar to Zehnder et al. [ZNT18], we indeed observe that the reflection scheme leads to better energy conservation, as can be seen in Fig. 4. The plot also shows that, unlike the implicit Euler and step-and-project approach, the energy evolution for the step-and-reflect scheme shows slight oscillations around a smoothly evolving curve. We conjecture that this behavior might be akin to the fluctuations observed in symplectic integration methods which, although non-monotonic, exhibit excellent long-term energy conservation [HLW06]. Using a second-order reflection scheme [NZT19] might further improve energy conservation.

We additionally experimented with BDF-2 variants of our end-of-step projection and reflection schemes for comparison. By directly replacing implicit Euler with BDF-2 in SAP, we recover the projection method by English and Bridson [EB08] to which we refer as BAP. In addition, we also implemented a reflection version of BAP, termed BAR, but noticed instability problems when using the BDF-2 formula for the mid-step velocity $\mathbf{v}^{1/2}$. As shown in Fig. 4, the energy curves of BDF-2 methods exhibit larger fluctuations, but show overall better energy conservation as expected. Finally, we also investigate the impact of step size on our SAR method. As can be seen in Fig. 5, SAR is robust even for large step sizes even though energy conservation deteriorates.

5. Results

We tested our method on a number of static and dynamic examples that we discuss below. In all cases, we use a square sheet of fabric with $0.5m$ side length initialized in the xz-plane and fixed at two corners. We use constant strain triangle (CST) elements [TPS09] and hinge-based elements [GHDS03] for modeling the in- and out-of-plane behavior of the material, respectively. We set the material thickness to $0.5mm$, its density to $200kg/m^3$, and use a Young's modulus $1800Pa$ and a Poisson's ratio of 0.41. Using these parameters, we evaluate our method for different deformation limits and different mesh resolutions.

Our method allows us to freely impose arbitrary bounds on stretching in any material direction. This, in turn, enables simulation of a large variety of materials, including quasi-inextensible isotropic materials, orthotropic fabrics with tight and loose bounds

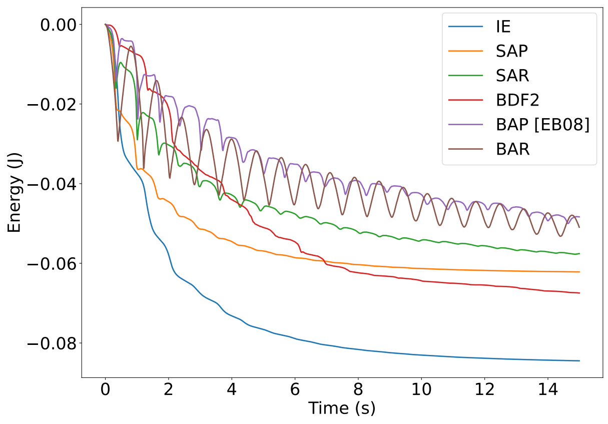

Figure 4: *Evolution of total energy over time for different methods on a swinging cloth example with 1% isotropic strain limits. Unconstrained implicit Euler (IE) loses energy more rapidly than our SOCP approach with end-of-step projection (SAP). Our step-and-reflect method (SAR) further improves energy conservation. For BDF-2 versions of projection (BAP) and reflection (BAR) methods, we obtain even lower numerical damping, but larger energy fluctuations.*

in different directions, as well as fully anisotropic weaves. As illustrated in Fig. 2, different combinations of deformation limits lead to different static drapes—and a similarly rich variety in dynamic behavior, as shown in the accompanying video. Another dynamic example illustrates the feasibility of combining our method with standard collision resolution algorithms (see Fig. 6).

During all these experiments, we observed that our SOCP-based method is extremely robust, yielding stable results for both static and dynamic examples that precisely satisfy all constraints, regardless of the deformation limits. We attribute this favorable behavior to the strict convexity of our formulation.

5.1. Performance

We analyze the performance of our method by measuring the average time spent inside the SOCP solver across 100 simulation steps of our swinging cloth example using different mesh resolutions. Note that our SAR method involves two projections, so we average across 200 SOCP solves. For reference, we also present the average time spent inside the Newton solver for implicit integration. As can be seen from Tab. 1, the timings for the SOCP solver and implicit Euler are comparable for small and moderate problem sizes. For larger problems, however, computation times increase much more rapidly for SOCP, taking roughly 8s per time step for 10,000 degrees of freedom and approximately 30,000 inequality constraints.

We further evaluated the impact of constraint tightness on computation times, but observed almost no effect when decreasing bounds from 20% to 0.1%. Similarly, we noticed only slight changes (approximately 20%) in computation time when increasing the number of constraints per element from 4 to 18.

To analyze the performance of our SOCP formulation, we compare to two common strain limiting methods: 1) solving a general nonlinear programming (NLP) problem (similar to [JLGF17]) and 2) using Gauss-Seidel (GS) to iteratively enforce constraints per element (similar to [TPS09]). We solve the SOCP using MOSEK 9.3 with default parameters, whereas the NLP is solved using Ipopt

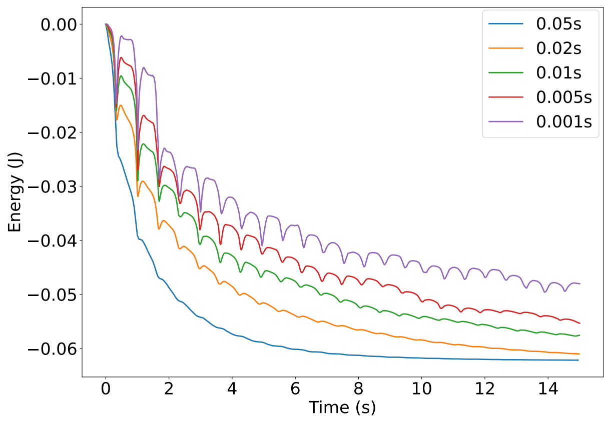

Figure 5: *Evolution of total energy over time for different step sizes on a swinging cloth example with 1% isotropic strain limits.*

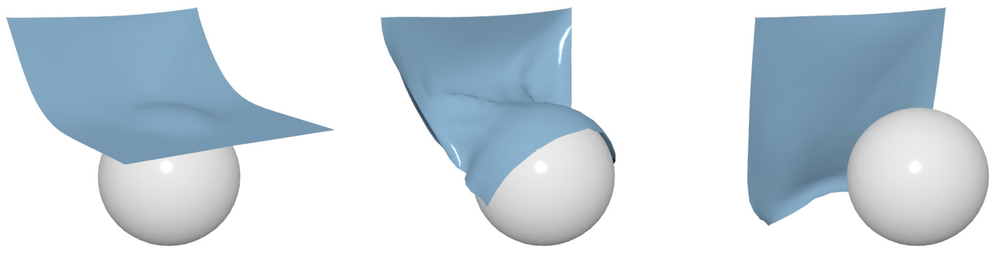

Figure 6: *Our SOCP-based strain limiting method can be integrated into standard collision resolution approaches. Here we use simple penalty functions that prevent intersection during time integration and enforce isotropic strain limits of* 0.1%.

Table 1: *Performance of our method compared to unconstrained implicit Euler (IE), general nonlinear programming (NLP), and Gauss-Seidel (GS) as a function of mesh resolution under isotropic strain limiting. Timings are averaged across 100 animation steps. The symbol* * *indicates that some steps did not converge whereas* $-$ *indicates solver failure.*

Mesh resolution		10×10	20×20	40×40	60×60
	IE	0.017s	0.064s	0.361s	1.318s
	Ours	0.108s	0.285s	1.289s	4.060s
1%	NLP	0.436s	$^*21.009s$	–	–
strain	GS	0.422s	$^*7.184s$	$^*54.725s$	$^*100.033s$
	Ours	0.107s	0.278s	1.242s	3.790s
10%	NLP	0.405s	$^*27.001s$	–	–
strain	GS	0.016s	0.238s	6.311s	$^*48.427s$

3.14 with default parameters. For GS, flag convergence once constraint violations drop below 1e-6 and set the maximum number of iterations to 1e5. As shown in Tab. 1, our method stands out over these two alternatives, especially for high-resolution meshes and tight deformation bounds. In addition, we can see that both NLP and GS may not converge or even fail to handle the problem for high-resolution meshes. In contrast, our SOCP method handles all of these cases robustly.

All examples were run on a machine with an Intel Core i9-7900X 3.3GHz processor and 32 GB of RAM.

6. Conclusion

We presented a new formulation for strain limiting based on second order cone programming. Unlike previous methods for globally-coupled strain limiting, our formulation is convex from the ground up and is guaranteed to have a single optimum. We have translated our quadratic inequality constraints into the conic form, allowing us to leverage powerful conic programming solvers.

While our SOCP excels at enforcing bounds on stretching, it cannot handle bounds on compression since the corresponding constraints are not convex. We argue that, for thin sheet materials such as textiles, this is a worthwhile compromise as compressions generally lead to immediate out-of-plane buckling.

Our initial results demonstrate the feasibility of our approach and indicate that, by setting arbitrary bounds on stretching in arbitrary directions, a large range of interesting material behaviors can be modeled. Our experiments further indicate that our SOCP-based approach is unconditionally robust, yielding stable and temporally smooth animations even for tight deformation bounds.

Computation times seem to be largely unaffected by constraint tightness and the number of constraints per element, but they grow rather rapidly with respect to mesh resolution. However, our method still outperforms conventional approaches by large margins, especially for high-resolution meshes and tight deformation bounds. While our formulation will automatically benefit from progress in SOCP solvers, multi-resolution approaches that enforce constraints on coarser levels than the simulation mesh might be a worthwhile direction for future exploration.

We have demonstrated that our method can be combined with a simple collision response approach. However, future work should investigate ways of integrating our method with advanced collision handling frameworks such as [NSO12] or [LKJ21].

While we have focused on in-plane behavior, it would also be interesting to develop quadratic formulations for bending deformation limits. Enforcing limits on bending deformation would enable simulating materials such as quasi-rigid chainmail.

Our results, both qualitative and quantitative, focus on simple geometries and simple test cases. While they constitute a proof-of-concept, we would like to confirm the promising behavior of our method on clothing examples with production-level complexity.

Finally, we believe that many other animation problems can benefit from conic programming reformulations. One particularly interesting case would be frictional contact modeling, for which SOCP methods have already shown their promise [BW07].

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments. This work was supported by the Discovery Grants Program and the Discovery Accelerator Awards program of the Natural Sciences and Engineering Research Council of Canada (NSERC). Computing equipment has been funded through an infrastructure grant from the Canada Foundation for Innovation (CFI). This work was also supported by the European Research Council (ERC) under the European Union's Horizon 2020 research, innovation program (grant agreement No. 866480), and the Swiss National Science Foundation through SNF project grant 200021_200644.

References

[AG03] ALIZADEH F., GOLDFARB D.: Second-order cone programming. *Mathematical programming 95*, 1 (2003), 3–51. 4

[ART03] ANDERSEN E. D., ROOS C., TERLAKY T.: On implementing a primal-dual interior-point method for conic quadratic optimization. *Mathematical Programming 95*, 2 (2003), 249–277. 1

[BFA02] BRIDSON R., FEDKIW R., ANDERSON J.: Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph. 21*, 3 (jul 2002), 594–603. 1, 2, 4

[BML*14] BOUAZIZ S., MARTIN S., LIU T., KAVAN L., PAULY M.: Projective dynamics: Fusing constraint projections for fast simulation. *ACM transactions on graphics (TOG) 33*, 4 (2014), 1–11. 2

[BMM15] BENDER J., MÜLLER M., MACKLIN M.: Position-based simulation methods in computer graphics. In *Eurographics (tutorials)* (2015), p. 8. 2

[BW07] BOYD S. P., WEGBREIT B.: Fast computation of optimal contact forces. *IEEE Transactions on Robotics 23*, 6 (2007), 1117–1132. 6

[DLL*18] DINEV D., LIU T., LI J., THOMASZEWSKI B., KAVAN L.: Fepr: Fast energy projection for real-time simulation of deformable objects. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 1–12. 2

[EB08] ENGLISH E., BRIDSON R.: Animating developable surfaces using nonconforming elements. In *ACM SIGGRAPH 2008 Papers* (New York, NY, USA, 2008), SIGGRAPH '08, Association for Computing Machinery. 2, 5

[GHDS03] GRINSPUN E., HIRANI A. N., DESBRUN M., SCHRÖDER P.: Discrete shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Goslar, DEU, 2003), SCA '03, Eurographics Association, p. 62–67. 5

[GHF*07] GOLDENTHAL R., HARMON D., FATTAL R., BERCOVIER M., GRINSPUN E.: Efficient simulation of inextensible cloth. *ACM Trans. Graph. 26*, 3 (jul 2007), 49–es. 1, 2, 3

[HCPO13] HERNANDEZ F., CIRIO G., PEREZ A. G., OTADUY M. A.: Anisotropic strain limiting. In *Proc. of Congreso Español de Informática Gráfica* (2013), vol. 2. 2

[HLW06] HAIRER E., LUBICH C., WANNER G.: *Geometric numerical integration*, second ed., vol. 31 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2006. Structure-preserving algorithms for ordinary differential equations. 4, 5

[JLGF17] JIN N., LU W., GENG Z., FEDKIW R. P.: Inequality cloth. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (New York, NY, USA, 2017), SCA '17, Association for Computing Machinery. 1, 2, 5

[LB15] LI Y., BARBIČ J.: Stable anisotropic materials. *IEEE transactions on visualization and computer graphics 21*, 10 (2015), 1129–1137. 2

[LKJ21] LI M., KAUFMAN D. M., JIANG C.: Codimensional incremental potential contact. *ACM Trans. Graph. 40*, 4 (jul 2021). 6

[LVBL98] LOBO M. S., VANDENBERGHE L., BOYD S., LEBRET H.: Applications of second-order cone programming. *Linear Algebra and its Applications 284*, 1 (1998), 193–228. International Linear Algebra Society (ILAS) Symposium on Fast Algorithms for Control, Signals and Image Processing. 4

[MBT*12] MIGUEL E., BRADLEY D., THOMASZEWSKI B., BICKEL B., MATUSIK W., OTADUY M. A., MARSCHNER S.: Data-driven estimation of cloth simulation models. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 519–528. 2

[MCKM15] MÜLLER M., CHENTANEZ N., KIM T.-Y., MACKLIN M.: Strain based dynamics. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Goslar, DEU, 2015), SCA '14, Eurographics Association, p. 149–157. 2

[NSO12] NARAIN R., SAMII A., O'BRIEN J. F.: Adaptive anisotropic remeshing for cloth simulation. *ACM Trans. Graph. 31*, 6 (nov 2012). 1, 6

[NZT19] NARAIN R., ZEHNDER J., THOMASZEWSKI B.: A second-order advection-reflection solver. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 2*, 2 (2019), 1–14. 5

[OBLN17] OVERBY M., BROWN G. E., LI J., NARAIN R.: Admm ⊆ projective dynamics: Fast simulation of hyperelastic models with dynamic constraints. *IEEE transactions on visualization and computer graphics 23*, 10 (2017), 2222–2234. 2

[P*95] PROVOT X., ET AL.: Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In *Graphics interface* (1995), Canadian Information Processing Society, pp. 147–147. 1, 2

[PCH*13] PEREZ A. G., CIRIO G., HERNANDEZ F., GARRE C., OTADUY M. A.: Strain limiting for soft finger contact simulation. In *2013 World Haptics Conference (WHC)* (2013), IEEE, pp. 79–84. 1, 2

[TPS09] THOMASZEWSKI B., PABST S., STRASSER W.: Continuum-based strain limiting. *Computer Graphics Forum 28*, 2 (2009), 569–576. 1, 2, 5

[WOR10] WANG H., O'BRIEN J., RAMAMOORTHI R.: Multi-resolution isotropic strain limiting. *ACM Trans. Graph. 29*, 6 (dec 2010). 2

[WOR11] WANG H., O'BRIEN J. F., RAMAMOORTHI R.: Data-driven elastic models for cloth: Modeling and measurement. *ACM Trans. Graph. 30*, 4 (jul 2011). 2

[XSZB15] XU H., SIN F., ZHU Y., BARBIČ J.: Nonlinear material design using principal stretches. *ACM Transactions on Graphics (TOG) 34*, 4 (2015), 1–11. 2

[ZNT18] ZEHNDER J., NARAIN R., THOMASZEWSKI B.: An advection-reflection solver for detail-preserving fluid simulation. *ACM Trans. Graph. 37*, 4 (jul 2018). 4

[ZPOD19] ZHANG J., PENG Y., OUYANG W., DENG B.: Accelerating admm for efficient simulation and optimization. *ACM Trans. Graph. 38*, 6 (nov 2019). 2

Appendix A

For the deformation gradient $\mathbf{F} \in \mathbb{R}^{3\times2}$, we compute it as

$$\mathbf{F} = \frac{d\mathbf{x}}{d\bar{\mathbf{x}}} = \frac{d\mathbf{x}\mathbf{N}(\bar{\mathbf{x}})}{d\bar{\mathbf{x}}} = \mathbf{x}\frac{d\mathbf{N}(\bar{\mathbf{x}})}{d\bar{\mathbf{x}}} = \mathbf{x}\mathbf{B}\,, \tag{13}$$

where $\mathbf{N}$ is the shape functions for the triangle. Therefore, we get the matrix $\mathbf{A}$ in Equation (3) as

$$\mathbf{A} = \begin{bmatrix} \bar{\mathbf{d}} & & \\ & \bar{\mathbf{d}} & \\ & & \bar{\mathbf{d}} \end{bmatrix}^T \begin{bmatrix} \mathbf{B}_1 & & \\ & \mathbf{B}_1 & \\ & & \mathbf{B}_1 \\ \mathbf{B}_2 & & \\ & \mathbf{B}_2 & \\ & & \mathbf{B}_2 \\ \mathbf{B}_3 & & \\ & \mathbf{B}_3 & \\ & & \mathbf{B}_3 \end{bmatrix}^T, \tag{14}$$

where $\mathbf{B}_i$ indicates the i-th row of matrix $\mathbf{B}$.