Efficient and Stable Approach to Elasticity and Collisions for Hair Animation

Galen Gornowicz
DreamWorks Animation

Silviu Borac
PDI/DreamWorks Animation

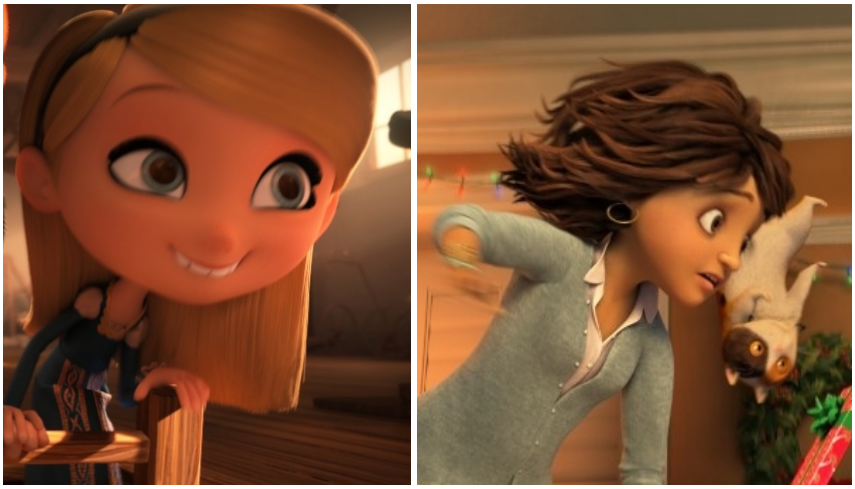

Figure 1: *Hair in motion.*

Abstract

We present a hybrid direct and projective iterative algorithm for simulating hair. The algorithm is fast, stable, visually accurate, uses a comprehensive elasticity model with a well defined continuum limit. The low resolution simulation is a good predictor of higher resolution results which we attribute to our novel treatment of bending and twisting forces. Timesteps may be as large a single step for each frame of 24 Hz animation. The treatment of collisions is implicit and does not introduce any artificial strain or forces.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling

Keywords: hair, rods, strands, collision, contact, friction, elasticity

1 Introduction

Physically-based simulation of hair has important applications for character animation and visual effects. Researchers have developed methods for its simulation based on a variety of hair models [Ward et al. 2007].

One of the models is Discrete Elastic Rods (DER) [Bergou et al. 2008] [Bergou et al. 2010], which is a discretization of the continuum model of [Kirchhoff 1859; Langer and Singer 1996], extended to stretchable rods.

Compared to other models, the existence of a well defined continuum limit for discrete elasticity models ensures that users of the system are able to set material parameters in an intuitive manner. In addition the model allows a completely general specification of the undeformed (stress-free) shape. Also, the model accurately captures the complexity of bending and twisting hair interactions.

1.1 Contributions

We present a method for the physically-based animation of colliding hair. The main contributions of our paper are:

A novel treatment of bending and twisting forces in a discrete elastic rod We introduce derivation of material curvatures (Section 3.2) that produces a force response that is nearly linear with respect to bending angle, greatly improving invariance with respect to discretization sampling. Furthermore our model ensures that the elastic energy Hessian is banded, symmetric, and semi-positive definite for any configuration of the discrete elastic rod, thereby doubling the efficiency of assembling and factorizing the linear system via Cholesky decomposition. Additionally, the number of Hessian terms is significantly lower than in prior work.

A novel and efficient algorithm for simulating hair's elasticity in conjunction with collision response and friction The algorithm (Section 6) does not alternate between solving for elasticity, and solving for collisions/friction. Instead, at each timestep the appropriate normal and friction forces are determined by solving the quadratic programming problem with an objective function determined by the internal dynamics and unilateral constraints arising from non-penetration and friction conditions. User defined hair-to-hair length constraints are solved along with the contacts.

Tightly integrating the treatment of internal dynamics and collisions removes a possible source of visual inaccuracy since there is no

need to limit strain rates during simulation.

A key ingredient of our algorithm is the banded Cholesky preconditioner. By itself the preconditioner is a direct solver for the internal dynamics. It is used in the combined elasticity and collisions solver as a building block which is essential for the efficiency of the algorithm.

2 Related Work

One of the earliest papers on hair simulation introduced the first mass-spring model for hair and demonstrated its inherent scalability [Rosenblum et al. 1991]. [Selle et al. 2008] extended the basic model by introducing additional springs to account for hair twist.

Multi-body hair models [Hadap 2006] use a system of rigid bodies connected by angular springs. The resulting articulated body dynamics problem is solved by applying Featherstone's forward dynamics algorithm [Featherstone 1987].

A hybrid between a multi body and mass-spring system was introduced in [Choe et al. 2005]. In this model rigid body links are connected with linear as well as twisting springs.

Whereas the first systems for hair animation [Ward et al. 2007] employed coarse level models for hair, more recent approaches use large scale, fiber level approaches [Selle et al. 2008; McAdams et al. 2009; Bertails-Descoubes et al. 2011; Daviet et al. 2011; Kaufman et al. 2014] which results in much more detailed animation that capture complex collective behavior.

A widely used strategy for simulating colliding soft bodies is to separate the treatment of collisions from that of internal forces. In the system introduced in [Bridson and Fedkiw 2002] the velocities of the soft body are estimated first from the internal dynamics alone and then the velocities are modified to account for the effect of collisions, resting contacts and friction, but processing contacts and collisions separately from internal forces can introduce artificial strain in the simulation results; a source of instability given the typically large stretch stiffness required in realistic hair simulation.

[Baraff and Witkin 1998] pioneered the combined treatment of collisions and elasticity by introducing a modified conjugate gradient method that can solve for internal dynamics subject to equality constraints. [Choe et al. 2005] applies this method to the simulation of hair.

One limitation of the modified conjugate gradient method is that it can process only equality constraints rather than the unilateral constraints required by collisions and friction. Substituting equality constraints for unilateral ones is problematic without a robust method for determining when constraints should be activated and released. Spurious large internal forces and sticking artifacts may occur when constraints are wrongly considered as active. A very informative illustration of these problems may be found in [Goldenthal 2010]. [Baraff and Witkin 1998] point out the difficulty of determining the activation state of constraints due to the combinatorial nature of the problem: the correct activation state of a constraint depends on the internal dynamics as well as the correct activation states of the other constraints. Effective solutions to the problem of determining exact Coulomb friction forces have been developed in [Bertails-Descoubes et al. 2011], which introduced a nonsmooth Newton algorithm, and in [Daviet et al. 2011], which combines a nonlinear complementarity solver with an analytical fail-safe solver to obtain a very robust system. [Kaufman et al. 2014] applies the approach in [Daviet et al. 2011] to large-scale elastic rod assemblies.

3 Hair Model

Linear elasticity theory assumes that strains under consideration are infinitesimal. The realm of hair deformation that we are interested in is the motion of natural hair, with its embedded curls, cascading around a character's head, reacting as the character walks, runs or jumps, or is blown by the wind. We expect the resulting deformations will have radius of curvature significantly greater than the diameter of hair and satisfy the assumptions of linear elasticity. Practically speaking, one would need to intentionally fold and pinch a hair in order for it to yield. (Note we're speaking of a single hair vs. the nonlinear behavior of a clump of hairs that will exhibit yielding due to slipping and locking of a clump of neighboring hairs).

While some choose to model hair as a continua [Hadap 2001, Bertails 2006], it's standard to formulate some discrete representation of straight spans and accumulate curvature at joints where the finite curvature is assumed to be distributed across the neighboring spans. We model individual hair strands using discrete elastic rods [Bergou et al. 2008; Bergou et al. 2010] with some modifications. Rods are allowed to stretch, bend and twist but not shear, so the continuum model is a generalization of the stretchable rods of Kirchhoff's theory [Kirchhoff 1859].

3.1 Discretization

Geometrically, a smooth elastic rod is represented as an adapted framed curve, which consists of a centerline $\mathbf{x}(s) \in \mathbf{R}^3$ and a material frame $[\mathbf{d}_0(s), \mathbf{d}_1(s), \mathbf{d}_2(s)] \in SO(3)$, where s is the coordinate along the centerline.

The motion of the hair can be understood in terms of the change in shape from the *undeformed* (stress-free) configuration $\overline{\mathbf{x}}(s),\ \overline{\mathbf{d}}_\alpha(s)$ to the deformed shape $\mathbf{x}(s,t),\ \mathbf{d}_\alpha(s,t)$ at the current time t.

The centerline of the rod is discretized as a sequence of $n+2$ vertices with positions $[\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{x}_{n+1}]$. To each edge $\mathbf{e}^j \stackrel{\text{def}}{=} \mathbf{x}_{j+1} - \mathbf{x}_j,\ 0 \le j \le n+1$ corresponds an orthonormal frame consisting of the material directors $\begin{bmatrix}\mathbf{d}_0^j & \mathbf{d}_1^j & \mathbf{d}_2^j\end{bmatrix}$. The frame is adapted to the discretized centerline, i.e., one of the unit vectors comprising the orthonormal frame is aligned with the tangent to the centerline:

$$\mathbf{d}_2 = \mathbf{t}^j \stackrel{\text{def}}{=} \frac{\mathbf{e}^j}{|\mathbf{e}^j|}, \tag{1}$$

where $|\mathbf{e}^j|$ is the edge's length.

Since the material frame must be both orthonormal and adapted to the discretized centerline, it follows that the time evolution of the material frame at an edge may be represented by just one scalar, τ^j, which is the angle that describes the incremental change in the orientation of the first material director $\mathbf{d}_0^j(t),\ t_0 \le t \le t_0 + h$ relative to its orientation at the beginning of the simulation step. The vector of generalized coordinates that describes the configuration of a hair is then: $\mathbf{q} = \begin{bmatrix}\mathbf{x}_0, \tau^0, \cdots, \mathbf{x}_n, \tau^n, \mathbf{x}_{n+1}\end{bmatrix} \in \mathbf{R}^{4n+7}$.

Note that in the interest of simplicity, we do not make use of any reference directors in addition to the material directors.

Discrete Curvature and Discrete Curvature Binormal Vector For an internal vertex i we define the discrete curvature as simply the turning angle Φ_i between two consecutive edges:

$$\Phi_i \stackrel{\text{def}}{=} \text{angle}(\mathbf{e}^{i-1}, \mathbf{e}^i),\ 1 \le i \le n. \tag{2}$$

This leads to the following expression for the discrete curvature binormal:

$$(\Phi\mathbf{b})_i = \Phi_i \frac{\mathbf{t}^{i-1} \times \mathbf{t}^i}{|\mathbf{t}^{i-1} \times \mathbf{t}^i|}. \tag{3}$$

This choice for the discrete curvature ensures that for circle-shaped elastic rods the bending energy is the same in the continuous and discrete cases, for all $n \geq 3$.

Discrete Material Curvatures When attempting to define the discrete material curvatures at an interior vertex we encounter the following problem: the discrete curvature κ_i is a vertex defined quantity whereas the material directors $\mathbf{d}_\alpha^{i-1}$, $\mathbf{d}_\alpha^i$, $\alpha \in \{0, 1\}$ are defined at the neighboring edges. Simple averaging of the neighboring edges' reference frames, as done in [Bergou et al. 2010], becomes increasingly inaccurate at higher bending and twisting angles, and results in a null vector when there is a 180° twist in the reference frame across the joint. To address this problem we obtain interpolated material directors at vertices by spherical interpolation of edge material directors:

$$\tilde{\mathbf{d}}_{\alpha,i} = \text{slerp}_{\frac{1}{2}} \left(\mathbf{P}^{\tilde{\mathbf{t}}_\mathbf{i}}_{\mathbf{t^{i-1}}} \mathbf{d}_\alpha^{\mathbf{i-1}}, \mathbf{P}^{\tilde{\mathbf{t}}_\mathbf{i}}_{\mathbf{t}^\mathbf{i}} \mathbf{d}_\alpha^{\mathbf{i}} \right), \tag{4}$$

where the parallel or zero-twist transport $P^{\mathbf{u}_2}_{\mathbf{u}_1}$ from a unit vector $\mathbf{u}_1$ to another unit vector $\mathbf{u}_2$ is the minimum angle rotation that aligns $\mathbf{u}_1$ with $\mathbf{u}_2$.

Finally, the discrete material curvatures are defined as $\phi_{\alpha,i} = \tilde{\mathbf{d}}_\alpha^i \cdot (\Phi\mathbf{b})_i$. And discrete twist at the vertex θ_i is defined as the angular difference of the adjacent material directors zero-twist transported onto the joint.

3.2 Bending And Twisting Forces

The Kirchhoff theory of rods assigns an elastic energy to an adapted framed curve from which internal elasticity forces are derived

$$F = -\frac{dE}{d\mathbf{q}}. \tag{5}$$

The total elastic energy consists of three terms corresponding to stretching, bending and twisting:

$$E = E_s + E_b + E_t. \tag{6}$$

Each of these energy functions is defined in terms of scalar functions that measure local deformations (*strains*). The derivation of stretching forces is straightforward (e.g. [Bergou et al. 2010]). We describe bending and twisting in detail below.

Derivation of Forces and Energy Hessian The internal bending and twisting energies for a joint centered around vertex i are defined as:

$$E_{b\alpha,i} = \frac{1}{2} \frac{k_{b\alpha,i}}{\bar{l}_i} (\phi_{\alpha,i} - \bar{\phi}_{\alpha,i})^2, \quad \alpha \in \{0, 1\}, \tag{7}$$

$$E_{t,i} = \frac{1}{2} \frac{k_t}{\bar{l}_i} (\theta_i - \bar{\theta}_i)^2. \tag{8}$$

Where the quantities k are stiffness constants which depend on the hair's material properties and geometry such as radius and eccentricity. Local rest length $\bar{l}_i$ is computed as the average of adjacent edge lengths.

Full differentiation of these energy expressions is an exhaustive exercise in derivation via the chain rule, and produces a relatively large number of complicated terms, the evaluation of which adds a significant cost to the total simulation time. More importantly, some of the resulting energy Hessian terms lead to an indefinite linear system which complicates their use in an implicit integration scheme. Numerical methods such as the generalized minimal residual method could overcome such terms, but at some cost in the subsequent squaring of the linear system's condition number would amplify convergence problems.

Identifying and separating indefinite terms is common and relatively easy for the stretching energy Hessian, that becomes indefinite under compressive loading. But a similar treatment proved much harder for the bending and twisting Hessian terms due to their greater number and mathematical complexity.

We found a simpler approach is to include only the dominant terms of the first derivative of the bend and twist energy. We exclude terms that arise due to elongation or contraction of the edge and justify the exclusion by noting that the forces arising from the stretching energy are much, much greater than the excluded terms from bend and twist. This greatly simplifies subsequent derivation of the Hessian, and all resulting terms have been found to be semi-positive definite.

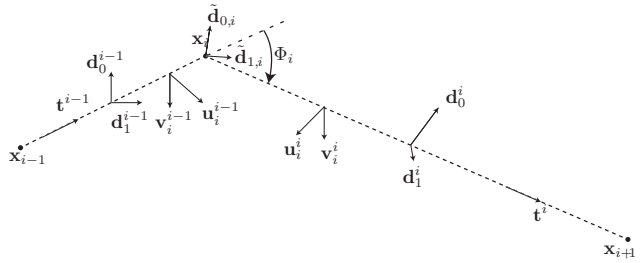

Figure 2: *Joint with labeled vector quantities.*

Differentiating the bending energy via the chain rule gives

$$\frac{\partial E_{b\alpha,i}}{\partial \mathbf{e}^j} = \frac{\partial E_{b\alpha,i}}{\partial \phi_{\alpha,i}} \frac{\partial \phi_{\alpha,i}}{\partial \mathbf{e}^j}. \tag{9}$$

The first term is simple to calculate, but the second is not. Instead we use vectors $\mathbf{u}_i^j$ and $\mathbf{v}_i^j$ of Figure 2 as approximations of the partial derivatives of the material curvature angles with respect to edge vectors,

$$\frac{\partial \phi_{0,i}}{\partial \mathbf{e}^j} \simeq \mathbf{u}_i^j = \left(\mathbf{e}^j \times \tilde{\mathbf{d}}_{0,i} \right) \frac{1}{|\mathbf{e}^j|^2} \beta_i, \quad j \in \{i-1, i\}, \tag{10}$$

$$\frac{\partial \phi_{1,i}}{\partial \mathbf{e}^j} \simeq \mathbf{v}_i^j = \left(\mathbf{e}^j \times \tilde{\mathbf{d}}_{1,i} \right) \frac{1}{|\mathbf{e}^j|^2} \beta_i. \tag{11}$$

We found the approximation introduces a small unbalanced torque at high bending angles. While it only becomes apparent during tests of free body (unattached) tightly curled hairs after many time steps. To compensate, we employ the factor β_i, a well behaved function of turning angle Φ_i, that was determined by applying conservation of angular momentum to a joint.

$$\beta(\Phi_i) = \frac{\Phi_i}{2} \left(\sin(\frac{\Phi_i}{2}) \right)^{-1}. \tag{12}$$

Our resulting force and Hessian expressions have relatively few components, making them easier to derive, verify, and ultimately less expensive to use. Some examples are shown in appendix [A].

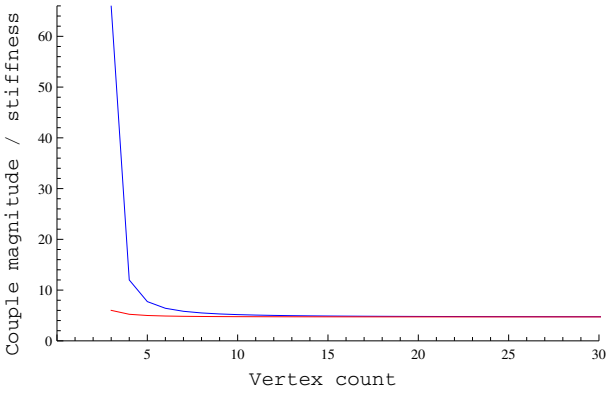

Figure 3: *Magnitude of the couple that must be applied at both ends of a naturally straight discretized elastic rod to keep it in static equilibrium while bent in the shape of a* 270° *arc for the model in [Bergou et al. 2010] (blue) and our model (red).*

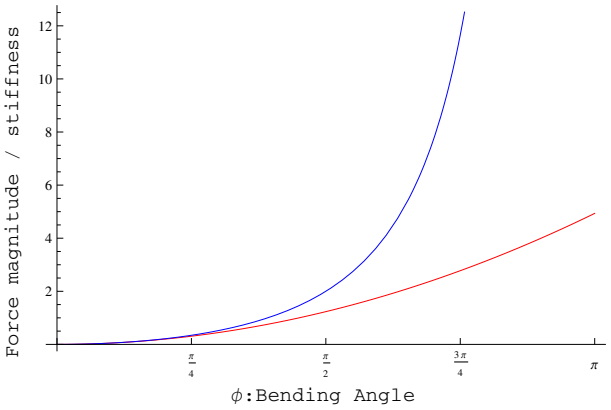

Figure 4: *Variation of bending force with turning angle in [Bergou et al. 2010] (blue) and our model (red).*

Analysis of Force Model Because a discretization is representing an underlying continua, the resulting force model should ideally be invariant with respect to choice of sampling. The invariance should hold within some reasonable limits such as satisfying a minimum sampling rate or having sufficient degrees of freedom where deformations are expected to take place. In our case, the sampling choice is the count and placement of vertices, and simulation efficiency benefits from the lowest possible resolution. This however focuses curvature onto fewer vertices, each of which will experience higher discrete turning angles for a given total deformation of a strand of hair. Thus we are particularly interested in the behavior of a model at low resolution discretization. We would expect a model's total force response to be consistent for any choice of sampling, and can thus evaluate correctness in terms of how well low and high resolution discretization match. We also interpret the high resolution behavior to be more correct because it better satisfies the assumptions of linear elasticity.

Figure 3 shows the magnitude of the couple that must be applied at both ends of a naturally straight discretized elastic rod to keep it in static equilibrium while bent in the shape of a 270°, for different discretization rates. We see that both our model (red) and the model of [Bergou et al. 2010] (blue) are self-consistent (and in fact consistent with each other) but only at higher resolution upper half of the vertex counts. In contrast, towards the lower end of the discretization, which again is of particular interest to practitioners, our force remains virtually constant even down to the minimum vertex count of three required to to represent a bent arc of hair. The other model exhibits a strong dependence on discretization, due to their model having a strongly non-linear force response with respect to discrete bending angle.

This behavior is further detailed in figure 4 which shows the bending force for the reference case of two unit edge lengths and a material director parallel to the binormal vs. the discrete bending angle. As the upper limit of 180° degrees for that single joint is approached, their force response experiences unbounded growth. We see that when starting from the elastic energy of Eq. 7, our model obtains a bending force that most importantly remains bounded over the entire domain of turning angles. The mild nonlinearity is due to the correction factor of Eq. 12 which we feel is admissible, particularly when modeling a finite thickness material such as a cluster of hair, however, evaluating the accuracy of the non-linearity is beyond the scope of this current work.

Damping forces are derived in a similar fashion, starting from the elastic energy expressions by differentiating with respect to both position and time.

3.3 Hair-to-hair Constraints

Our hair animation system is used primarily for stylized hair. In this context it is important to provide several ways for art directing hair-to-hair sticking and separation. Soft constraints between vertices on different guide hairs or belonging to the same guide hair may be used in conjunction with Coulomb friction.

Constraints are modeled as damped springs that are biphasic in the sense that the compression stiffness and damping may be different from the stretching stiffness and damping. Optionally we allow the stretching stiffness to diminish when the distance between the constrained points increases, to model the smaller resistance to separation of a hair clump with increased separation distance. In terms of implementation, hair-to-hair constraints are handled together with contact constraints, see Section 6.

4 Implicit Integration

Let $\mathbf{q}^t = \left(\mathbf{x}_0, \tau^0, \cdots, \mathbf{x}_n, \tau^n, \mathbf{x}_{n+1}\right)$ and $\dot{\mathbf{q}}^t$ be the generalized coordinates and velocities, respectively, at time step t. The corresponding quantities at the end of the previous time step are $\mathbf{q}^{t-1}, \dot{\mathbf{q}}^{t-1}$.

For integrating over time we use either implicit Euler [Baraff and Witkin 1998] or implicit midpoint (see, e.g. [Hairer et al. 2006]). We eliminate $\dot{\mathbf{q}}_t$ to directly relate the change in positions and angles to forces and the force Jacobian. For implicit Euler:

$$\frac{\mathbf{M}}{h^2}\left(\mathbf{q}^t - \mathbf{q}^{t-1}\right) = \frac{\mathbf{M}}{h}\dot{\mathbf{q}}^{t-1} + \mathbf{f}(\mathbf{q}_t, \dot{\mathbf{q}}_t) \tag{13}$$

where h is the time step and $\mathbf{f}$ are the internal elasticity, material damping and gravity forces, and $\mathbf{M}$ is the mass matrix.

Since $\mathbf{f}$ depends non-linearly on $\mathbf{q}$ and $\dot{\mathbf{q}}$ we use several Newton iterations for each timestep. Let $\mathbf{q}_i^t$ and $\dot{\mathbf{q}}_i^t = \frac{1}{h}\left(\mathbf{q}_i^t - \mathbf{q}^{t-1}\right)$ be i-th iteration estimates for $\mathbf{q}_t$ and $\dot{\mathbf{q}}_t$, respectively. As starting conditions for the iterative process we set $\left(\mathbf{q}_0^t, \dot{\mathbf{q}}_0^t\right) = \left(\mathbf{q}^{t-1}, \dot{\mathbf{q}}^{t-1}\right)$. Using a Taylor expansion of $\mathbf{f}$ we obtain the following linear system

of equations:

$$\left(\frac{\mathbf{M}}{h^2} - \frac{\partial \mathbf{f}}{\partial \mathbf{q}} - \frac{1}{h}\frac{\partial \mathbf{f}}{\partial \dot{\mathbf{q}}}\right)\Delta \mathbf{q} = \\ -\frac{\mathbf{M}}{h^2}\left(\mathbf{q}_{i-1}^t - \mathbf{q}^{t-1}\right) + \frac{\mathbf{M}}{h}\dot{\mathbf{q}}^t + \mathbf{f} + \frac{\partial \mathbf{f}}{\partial \dot{\mathbf{q}}}\dot{\mathbf{q}}^t \qquad (14)$$

where $\Delta \mathbf{q} = \left(\mathbf{q}_i^t - \mathbf{q}_{i-1}^t\right)$ and the forces and their derivatives are evaluated at $\left(\mathbf{q}_{i-1}^t,\ \dot{\mathbf{q}}_{i-1}^t\right)$. The corresponding equation for implicit midpoint is:

$$\left(\frac{2\mathbf{M}}{h^2} - \frac{1}{2}\frac{\partial \mathbf{f}}{\partial \mathbf{q}} - \frac{1}{h}\frac{\partial \mathbf{f}}{\partial \dot{\mathbf{q}}}\right)\Delta \mathbf{q} = \\ -\frac{2\mathbf{M}}{h^2}\left(\mathbf{q}_{i-1}^t - \mathbf{q}^{t-1}\right) + \frac{2\mathbf{M}}{h}\dot{\mathbf{q}}^t + \mathbf{f} + \frac{\partial \mathbf{f}}{\partial \dot{\mathbf{q}}}\dot{\mathbf{q}}^t \qquad (15)$$

Collision detection and response (Section 6) as well as constraint satisfaction is integrated within the implicit solver without alternating between them and the elasticity solve. Algorithm 1 is used for advancing the simulation.

5 Banded Cholesky Factorization

A linear system of equations with a symmetric and positive definite matrix $\mathbf{A}$ can be solved in a numerically stable and efficient way using the Cholesky factorization $\mathbf{A} = \mathbf{L}\mathbf{L}^T$. The stability of the Cholesky factorization follows from the fact that the elements of $\mathbf{L}$ do not grow since they are tightly bounded by the diagonal elements of $\mathbf{A}$: $\mathbf{L}_{ij}^2 \leq \mathbf{A}_{ii}$, [Golub and Loan 1996]. Consequently, one can avoid pivoting and its large number of memory-write operations that may severely impact the run time efficiency of a system that relies heavily on memory caches and multi-threading.

Note that we compute a *complete* Cholesky factorization. We rely on the fact that the Cholesky factors of a banded matrix are themselves banded and have the same bandwidth. Here we use the standard definitions for "banded matrix" (see [Golub96]): entries within a row that are further from the diagonal than the bandwidth are all zero. In our case the bandwidth is a small constant (11) regardless of the size of the problem

6 Collision

We detect and resolve hair to hair and hair to surfaces collisions. We employ both static proximity detection with offset as well as continuous collision detection without offset. Both detection types are accelerated using an axis-aligned bounding box tree (AABB) approach [van den Bergen et al. 1998]. The result of collision detection is a set of candidate contact constraints.

Before we consider solving contacts as part of our full hybrid solver, let us assume at first that we have only contact constraints and no elastic forces. At the time of detection we compute the following quantities for each contact: penetration depth p, normal restitution vector $\mathbf{n}$, and points of contact on either primitive expressed as barycentric weights β_a and β_b. We construct the following linear system for the normal forces required to resolve the constraints over the given timestep (see [Baraff 1996] for the equality constraints case):

$$\frac{h^2}{2}\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\lambda = \mathbf{p}, \qquad \lambda_i \geq 0. \qquad (16)$$

Once solved for lambda, the normal force corresponding to constraint j is computed as $\mathbf{f}_{nj} = \lambda_j \mathbf{n}_j$. Matrix $\mathbf{J}$ is the sparse $m \times 3n$ matrix encoding the m constraint's inter-connectivity. A 1×3 blockwise element definition of $\mathbf{J}$ corresponding to constraint j and vertex i is:

$$\mathbf{J}_{ji} = \mathbf{n}_j\left(\beta_{aji} - \beta_{bji}\right). \qquad (17)$$

The linear complementarity problem of equations (16) is solved using the Projected Gauss-Seidel (PGS) method [Catto 2005], which determines the active set of positive (pushing) non-zero normal forces required to satisfy the constraints. However, the formulation does not include internal elasticity forces, so the result of simply

Algorithm 1 Single timestep of Hybrid Solver

$\mathbf{O}, \mathbf{T}, S$ are the set of objects, the set of tasks and the number of parallelization pools (Section 8), max_newton, max_cbs, max_lcp are the number of Newton, Cholesky backsolve and LCP iterations (Section 6), and $\mathbf{A}, \mathbf{b}$ are the matrix and RHS of Equation (14).

for o in KinematicObjects($\mathbf{O}$) in parallel **do**
  $\mathbf{q}^t \leftarrow$ UpdatePositions()
  $\text{bbt}_o \leftarrow$ UpdateAABBTree($\mathbf{q}^t$)
end for o
for i in $\{1, \ldots,$ max_newton$\}$ **do**
  for o in DynamicObjects($\mathbf{O}$) in parallel **do**
    $\mathbf{A} \leftarrow \left(\frac{\mathbf{M}}{h^2} - \frac{\partial \mathbf{f}}{\partial \mathbf{q}} - \frac{1}{h}\frac{\partial \mathbf{f}}{\partial \dot{\mathbf{q}}}\right)$
    $\mathbf{b} \leftarrow -\frac{\mathbf{M}}{h^2}\left(\mathbf{q}_{i-1}^t - \mathbf{q}^{t-1}\right) + \frac{\mathbf{M}}{h}\dot{\mathbf{q}}^t + \mathbf{f} + \frac{\partial \mathbf{f}}{\partial \dot{\mathbf{q}}}\dot{\mathbf{q}}^t$
    $\mathbf{L} \leftarrow$ CholeskyFactorize($\mathbf{A}$)
    $\text{bbt}_o \leftarrow$ UpdateAABBTree(o)
  end for o
  $\{\mathbf{K}_1, \cdots, \mathbf{K}_{|\mathbf{O}|}\} \leftarrow$ ProximityDetection($\mathbf{O}, \mathbf{q}$)
  for o in $\mathbf{O}$ in parallel **do**
    for k in $\mathbf{K}_o$ **do**
      $\{p, \mathbf{n}, \beta\} \leftarrow$ BuildConstraint($\mathbf{q}$)
      $\lambda \leftarrow 0, \mathbf{F}_f \leftarrow 0$
    end for k
  end for o
  $\mathbf{T} \leftarrow$ FindInteractingObjectPairs($\mathbf{K}$)
  $\{\mathbf{T}_1, \cdots, \mathbf{T}_S\} \leftarrow$ ParallelPartitionTasks($\mathbf{T}$)
  for cbsitr in $\{1, \ldots,$ max_cbs$\}$ **do**
    for lcpitr in $\{1, \ldots,$ max_lcp$\}$ **do**
      for s in $\{1, \ldots, S\}$ **do**
        for task in $\mathbf{T}_s$ in parallel **do**
          $o \leftarrow$ TaskObject(task)
          for k in $\mathbf{K}_o$ **do**
            if IsIntraObjectContact(k) **then**
              $\lambda_k \leftarrow$ GaussSeidelUpdate($\mathbf{q}, \lambda$)
            else
              $\lambda_k \leftarrow$ JacobiUpdate($\mathbf{q}, \lambda$)
            end if
            $\mathbf{F}_f \leftarrow$ EvalFriction(λ)
          end for k
        end for task
      end for s
    end for lcpitr
    $\Delta \mathbf{q} \leftarrow$ CholeskyBackSolve($\mathbf{L}, \mathbf{b}, \lambda, \mathbf{F}_f$)
  end for cbsitr
  $\mathbf{q}_i^t \leftarrow \mathbf{q}_{i-1}^t + \Delta \mathbf{q}$
end for i
$\mathbf{q}^t \leftarrow \mathbf{q}^t_{\text{max\_newton}}$
$\dot{\mathbf{q}}^t \leftarrow \frac{1}{h}(\mathbf{q}^t - \mathbf{q}^{t-1})$
return $(\mathbf{q}^t, \dot{\mathbf{q}}^t)$

solving the equation is *local* to the neighborhood of the constraint.

After the normal force has been found, the associated friction force due to the contact is computed as:

$$f_f = \min\left(\frac{2}{h^2 w}\Delta\mathbf{x},\ \mu\lambda\frac{\Delta\mathbf{x}}{|\Delta\mathbf{x}|}\right), \tag{18}$$

where $\Delta\mathbf{x}$ is the relative motion between the two points of the contact, and $w = \frac{1}{m_a} + \frac{1}{m_b}$ is the sum of the inverse local masses involved in the contact. Hence the first option of the minimum operation represents the force required to completely arrest the motion between the two points of contact. The second option represents the maximum available friction force with coefficient of friction μ, which is either the kinetic coefficient μ_k if the magnitude of relative motion exceeds some small distance, otherwise we use the (larger) coefficient of static friction μ_s.

7 Hybrid Solver

We developed a hybrid direct and projective iterative solver to simultaneously solve normal and friction forces for all candidate contacts constraints as well as forces associated with user defined constraints (hereafter both shall be referred to simply as the set of constraints). Specifically we use Cholesky factorization and backsolve, and a combination of Projected Gauss-Seidel (PGS) and Projected Jacobi (PJ) solvers. While developed independently and in use for a number of years on production feature films, our method does have a number of similarities with the ADONIS algorithm of [Kaufman et al. 2014] as well as some important differences which we will point out.

Our hybrid solver is an iterative method for solving the following quadratic programming problem arising from evolving the dynamical state subject to both internal elasticity forces and contact constraints:

$$\text{Minimize with respect to } \Delta\mathbf{q}\text{:}\quad \frac{1}{2}\Delta\mathbf{q}^T\mathbf{A}\Delta\mathbf{q} - \mathbf{b}^T\Delta\mathbf{q} \tag{19}$$

$$\text{subject to:}\quad \mathbf{C}\Delta\mathbf{q} \le \mathbf{c}, \tag{20}$$

where $\mathbf{A}$ and $\mathbf{b}$ are, respectively, the LHS and RHS of Eq. (14) and $(\mathbf{C},\ \mathbf{c})$ express the non-penetration constraints as well as Coulomb-type constraints on the magnitudes of friction forces.

Our solver as described in Algorithm 1 requires a single Cholesky factorization of the elasticity matrix per Newton iteration. However, prior to applying the Cholesky backsolve, we first solve the constraint equations for λ then compute the corresponding constraint forces. Other than our use of Cholesky factorization, a major difference from [Kaufman et al. 2014] is that we couple the solution of our iterative solve not by positional displacements, but by *adding the forces to the RHS of the Cholesky backsolve* to find a new estimate of the state vector $(\mathbf{q}^t, \dot{\mathbf{q}}^t)$ for this iteration.

The advantage of this approach is that the internal dynamics of the hair remain paramount and we do not suffer undue strain of the hair. There is of course a limitation in that the constraints may not be fully resolved when termination is reached. In fact, we experimented with various means of enforcing constraint satisfaction such as a final PGS pass, warm-starting the PGS solve via tracking contacts between iterations. While all of those helped to ensure constraints were satisfied, in practice they all suffer the same downside: simulation jitter or outright failure in the presence of unsolvable sets of constraints such as those that may arise from self-intersecting collision geometry that one often sees in real world character simulations.

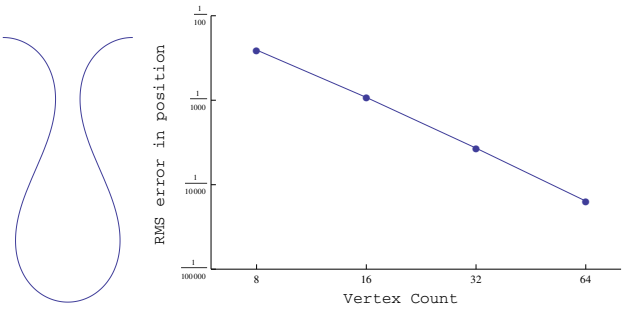

Figure 5: *The root mean square (left) of the distance between the simulation result and the analytically computed elastica curve (right), normalized to a rod of unit length.*

8 Parallelization

The inclusion of hair to hair collisions and constraints make the parallelization of the algorithm difficult. Without them the system parallelization would be trivial since each hair would depend only on its internal elasticity and collision against solid objects. We group neighboring hairs together into an object of an appropriate size to balance overhead and workloads, which in our case was found to be around $1,000$ vertices per object. The collision detection and state update can be computed independently for each object.

Inter-object and intra-object collisions are detected once per Newton iteration, while user constraints are given as inputs to the system. From the list of constraints and contacts we generate a list of all interacting object pairs. We then schedule using graph-color partitioning to create a list of pools that will be run sequentially. Each pool is a fully parallelizable list of task pairs. An object will appear no more than once in each pool. A typical large hair simulation might have somewhere on the order of 10K contacts. Due to the sparse nature of contacts and constraints, a contact typically connects to a small set of neighbors. Consequently the number of pools is relatively small, typically on the order of 10. Each pool can be run in a fully parallel loop. All threads must synchronize prior to beginning work on the next pool. To avoid stalling, the order of object pair tasks is sorted in descending cost which is estimated as the number of individual contacts or constraints between the two objects.

The cost of scheduling is minimal, less than 1% of the overall budget, and only needs to be recomputed after new collisions are detected.

9 Results and Discussion

Elastica curve To validate our bending model we have compared the output from the simulator with a known analytical solution to the problem of static equilibrium of a continuous elastic rod clamped at both ends (see Figure 5). The solution belongs to the family of inflectional elastica curves, which have closed forms expressible in terms of Jacobi elliptic functions and are labeled by a continuous parameter m (see Section 13 in [Levien 2008] for details). In our case $m = \frac{7}{10}$.

Helical buckling The interaction between bending and twisting is the cause of a number of instability phenomena. One of these is helical buckling where a naturally straight rod clamped at both ends and twisted maintains its straight shape for as long as the amount of twist is smaller than a threshold value and buckles into the shape of

Figure 6: *Top: Horse mane and tail. Down: Part of the mane is constrained to follow the character's hand.*

a helix with variable radius past that threshold. Our model reproduces the correct behavior.

Collisions with surfaces In an animation scene, hair interacts with character's skin, clothing and other objects in the environment. These objects are kinematic in the sense that, from the perspective of hair simulation, their motion and shape deformations are prescribed procedurally. In the scene shown in Figure 1 the collider objects for hair are the character's skin, dress, lute and floor boards. Note that collisions are handled robustly although the character design is such that the neck, arms and body are all thin relative to the size of the hair style.

Hair-to-hair collisions Hair-to-hair collisions with friction are used to prevent volume loss and to model the sliding effects between different layers of hair. Collisions are detected between different strands as well as within the same hair. To ensure a robust and stable collision response we use both vertex-edge and edge-edge proximity detection to generate contacts.

Hair-to-hair constraints To generate the hair-to-hair constraints (Section 3.3) we run proximity detection on the rest shape of the hair. We set a maximum on the number of constraints a hair vertex may belong to, the result is a network of constraints which in this case purposely hold all the hairs in a cohesive mass. The stiffness and damping parameters of constraints are usually set according to the position of the constraint within the hair style, though as a rule of thumb we found damping rates around 10% of the elasticity stiffness give a natural look.

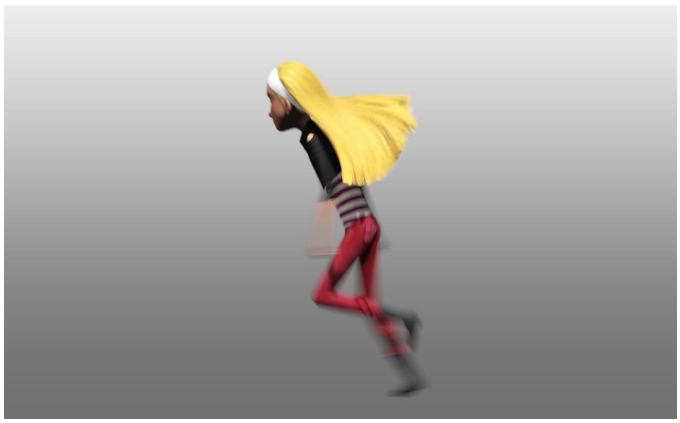

Figure 7: *Long hair style scene.*

Performance and scaling Table 1 details the performance of our system in average milliseconds per frame for a variety of cases including those depicted in Figures 1, 6, and 7. These setups encompass a large range of vertices, constraints, and average contacts. Given our use of a fixed iteration count, and the use of a direct solver on the dense banded linear system, we see the expected average running time is $\mathrm{O}(n+m)$ where n is the number of vertices, and m is the number of constraints and contacts.

Table 2 shows the performance of our system for a range of workloads and threads. In the case of a single thread, performance scales roughly linearly with increase of vertices, this is even despite a nearly quadratic increase in the number of contacts. However, as the number of threads increases this efficiency diminishes. We attribute this to the difficulty of parallelizing the constraint resolution, but note that performance decreases to 75% scaling efficiency in the case of 4 processors, and is still above 50% when utilizing 12 processors.

Parallelization efficiency is best in the case with the heaviest workload ($47K$ vertices), it ranges from 68% for four processors, and drops to 37% on 12 processors. We attribute the difference between ideal threading efficiency to the non-trivial task of threading so many constraints and contacts which in all test cases outnumber the vertex counts by an order of magnitude.

Limitation The main limitation of our system is that the fixed (user selectable) iteration count means we may not fully resolve collisions in challenging cases. Conversely when there are no collisions or constraints cycles are wasted since the the hair converges in one iteration due to the direct solve.

Conclusions

We have presented an efficient and novel system for the simulation of elasticity in conjunction with collision response, friction, and user defined constraints. Our algorithm is a hybrid solver with the advantage of a direct solver, it allows us to efficiently handle hairs with high vertex counts. At the same time we also retain the advantage of a projective iterative solver enabling the processing of high contact and constraint counts. The simulator has been used on multiple feature film production for various types of hair.

Model	Hairs	Hair Points	Colliders Points	Constraints	Average Contacts	Average Frame	Slowest Frame	Average collision detection time
Penny	300	5,918	15,795	38,432	17,184	0.352 s	0.471 s	0.108 s
Penny (8x density)	300	47,456	15,795	95,397	423,937	1.547 s	2.180 s	0.407 s
Horse mane	243	2,430	10,534	6,015	1,118	0.109 s	0.123 s	0.37 s
Horse tail	45	450	10,534	4,485	433	0.082 s	0.091 s	0.33 s
Long hair style	256	6,400	6,850	19,060	30,517	0.304 s	0.482 s	0.111 s
Hair ball	32,030	960,900	72,000	0	273,216	204.3 s	688 s	31.04 s

Table 1: *Timing results for simulations. Datasets Penny, Penny8x, Horse mane, Horse tail, Long hair used 8 substeps per frame and* $\texttt{cbs\_max} = 3$*. The Hair ball dataset used 4 substeps per frame and* $\texttt{cbs\_max} = 12$*. For all data sets we set* $\texttt{lcp\_max} = 3$, $\texttt{newton\_max} = 2$*. 'Collider Points' is the total vertex count over all collider objects. Results were measured on a multi-threaded implementation running on a 3.07GHz machine with 12 cores.*

Model	Hair Points	Constraints	Average Contacts	1	4	8	12
Penny	5,918	38,432	17,184	739 ms	423 ms	392 ms	352 ms
Penny (4x density)	23,759	48,059	123,221	2,657 ms	1,067 ms	814 ms	766 ms
Penny (8x density)	47,456	95,397	423,937	6,882 ms	2,519 ms	1,824 ms	1,547 ms

Table 2: *Parallelization performance. Results were measured on a multi-threaded implementation running on a 3.07GHz machine with 12 cores.*

Acknowledgements

The authors are most grateful to Brian Missey, Aaron Conover, and Kris Campbell for setting up production shots and to David Hill, Ron Henderson, and Andrew Pearce for their helpful suggestions while reviewing early drafts of this work.

References

BARAFF, D., AND WITKIN, A. 1998. Large steps in cloth simulation. In *Proceedings of SIGGRAPH 1998*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, ACM.

BARAFF, D. 1996. Linear-time dynamics using Lagrange multipliers. In *Proceedings of SIGGRAPH 1996*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, ACM.

BERGOU, M., WARDETZKY, M., ROBINSON, S., AUDOLY, B., AND GRINSPUN, E. 2008. Discrete elastic rods. *SIGGRAPH (ACM Transactions on Graphics) 27*, 3 (Aug.).

BERGOU, M., AUDOLY, B., VOUGA, E., WARDETZKY, M., AND GRINSPUN, E. 2010. Discrete viscous threads. *SIGGRAPH (ACM Transactions on Graphics) 29*, 4 (July).

BERTAILS-DESCOUBES, F., CADOUX, F., DAVIET, G., AND ACARY, V. 2011. A nonsmooth newton solver for capturing exact coulomb friction in fiber assemblies. *ACM Trans. Graph. 30*, 1, 6.

BRIDSON, R., AND FEDKIW, R. 2002. Robust treatment of collisions, contact and friction for cloth animation. *SIGGRAPH (ACM Transactions on Graphics) 21*, 3 (July).

CATTO, E. 2005. Iterative dynamics with temporal coherence. In *Proc. of the 2005 Game Developers Conference*.

CHOE, B., CHOI, M. G., AND KO, H. 2005. Simulating complex hair with robust collision handling. In *Proc. of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 153–160.

DAVIET, G., BERTAILS-DESCOUBES, F., AND BOISSIEUX, L. 2011. A hybrid iterative solver for robustly capturing coulomb friction in hair dynamics. *ACM Trans. Graph. 30*, 6, 139.

FEATHERSTONE, R. 1987. *Robot Dynamics Algorithms*. Kluwer, Boston, MA.

GOLDENTHAL, A. R. 2010. *Implicit Treatment of Constraints for Cloth Simulation*. PhD thesis, Hebrew University.

GOLUB, G. H., AND LOAN, C. F. V. 1996. *Matrix Computations*. The Johns Hopkins University Press.

HADAP, S. 2006. Oriented strands — dynamics of stiff multibody system. In *ACM SIGGRAPH/Eurographics symposium on Computer Animation*.

HAIRER, E., LUBICH, C., AND WANNER, G. 2006. *Geometric Numerical Integration*. Springer.

KAUFMAN, D. M., TAMSTORF, R., SMITH, B., AUBRY, J.-M., AND GRINSPUN, E. 2014. Adaptive nonlinearity for collisions in complex rod assemblies. *ACM Trans. on Graphics (SIGGRAPH 2014)*.

KIRCHHOFF, G. 1859. Ueber das Gleichgewicht und die Bewegung eines unendlich dunnen elastichen Stabes. *Journal fur die reine und angewandte Mathematik 56*.

LANGER, J., AND SINGER, D. A. 1996. Lagrangian aspects of the Kirchhoff elastic rod. *SIAM Review 38*, 4.

LEVIEN, R. 2008. The elastica: a mathematical history. Tech. rep., University of California, Berkeley.

MCADAMS, A., SELLE, A., WARD, K., SIFAKIS, E., AND TERAN, J. 2009. Detail preserving continuum simulation of straight hair. *ACM Trans. Graph. 28*, 3.

ROSENBLUM, R. E., CARLSON, W. E., AND III, E. T. 1991. Simulating the structure and dynamics of human hair: modelling, rendering and animation. *J. Vis. and Comput. Anim. 2*, 4, 141–148.

SELLE, A., LENTINE, M., AND FEDKIW, R. 2008. A mass spring model for hair simulation. *SIGGRAPH (ACM Transactions on Graphics) 27*, 3 (Aug.).

VAN DEN BERGEN, G., VAN, G., AND BERGEN, D. 1998. Efficient collision detection of complex deformable models using aabb trees. *J. Graphics Tools 2*.

WARD, K., BERTAILS, F., KIM, T., MARSCHNER, S. R., CANI, M., AND LIN, M. C. 2007. A survey on hair modeling: Styling, simulation, and rendering. *IEEE Trans. Vis. Comput. Graph. 13*, 2, 213–234.

A Bending and Twisting Derivatives

The formulation for internal elastic energy chosen in this paper results in compact expressions with relatively few terms for the first and second derivatives that are required in the implicit solver. Below are select examples for derivatives used in the force and torsion expressions that arise from one component bending energy:

$$\frac{\partial E_{b0}}{\partial \mathbf{e}^j} = \sum_{i=j}^{j+1} (-1)^{i-j-1} \frac{k_{b0,i}}{\bar{l}_i} (\phi_{0,i} - \overline{\phi}_{0,i}) \mathbf{u}_i^j,$$

$$\frac{\partial E_{b0}}{\partial \tau^j} = \sum_{i=j}^{j+1} \frac{k_{b0,i}}{\bar{l}_i} (\phi_{0,i} - \overline{\phi}_{0,i}) \frac{\phi_{1,i}}{2}.$$

Other terms are similar in form, such as for force and torsion due to twist energy:

$$\frac{\partial E_t}{\partial \mathbf{e}^j} = \sum_{i=j}^{j+1} (-1)^{i-j-1} \frac{k_{t,i}}{\bar{l}_i} (\theta_i - \overline{\theta_i}) \left(\tan \frac{\Phi}{2} \right) \frac{1}{\Phi |\mathbf{e}^j|} (\Phi \mathbf{b})_i,$$

$$\frac{\partial E_t}{\partial \tau^j} = \sum_{i=j}^{j+1} \frac{k_{t,i}}{\bar{l}_i} (\theta_i - \overline{\theta_i}).$$

The energy Hessian terms are also relative simple compared to derivatives used in other approaches [Bergou et al. 2010]. Some example terms are shown below, for simplicity and brevity the indexing and summations have been omitted:

$$\frac{\partial^2 E_{b0}}{\partial \mathbf{e} \partial \mathbf{e}} = \frac{k_{b0}}{\bar{l}} \mathbf{u} \otimes \mathbf{u}, \qquad \frac{\partial^2 E_t}{\partial \mathbf{e} \partial \mathbf{e}} = \frac{k_t}{\bar{l}} \frac{1}{|\mathbf{e}|^2} (\Phi \mathbf{b}) \otimes (\Phi \mathbf{b}),$$

$$\frac{\partial^2 E_{b0}}{\partial \mathbf{e} \partial \tau} = \frac{k_{b0}}{\bar{l}} \frac{-\phi_1}{2} \mathbf{u}, \qquad \frac{\partial^2 E_t}{\partial \mathbf{e} \partial \tau} = \frac{k_t}{\bar{l}} \frac{1}{|\mathbf{e}|} (\Phi \mathbf{b}),$$

$$\frac{\partial^2 E_{b0}}{\partial \tau \partial \tau} = \frac{k_{b0}}{\bar{l}} \frac{\phi_1^2}{4}, \qquad \frac{\partial^2 E_t}{\partial \tau \partial \tau} = \frac{k_t}{\bar{l}}.$$