Stable Constrained Dynamics

Maxime Tournier [4,1,2] Matthieu Nesme [1,3] Benjamin Gilles [2,1] François Faure [5,3,1]

[1] INRIA [2] LIRMM-CNRS [3] LJK-CNRS [4] RIKEN BSI-BTCC [5] Univ. Grenoble

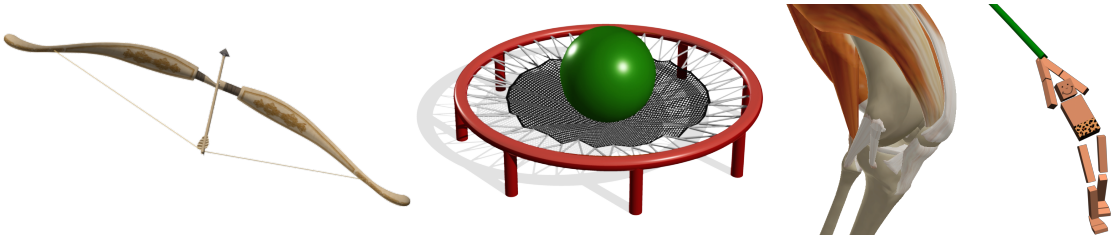


Figure 1: *Our method improves stability and step size for the simulation of constraint-based objects subject to high tensile forces, isolated or coupled with other types of objects. Bow: stiff 3D frame, 1D inextensible string, rigid arrow ; Trampoline: soft lateral springs, inextensible textile ; Knee: complex assembly of rigid bodies and stiff unilateral springs ; Ragdoll: rigid body assembly.*

Abstract

We present a unification of the two main approaches to simulate deformable solids, namely elasticity and constraints. Elasticity accurately handles soft to moderately stiff objects, but becomes numerically hard as stiffness increases. Constraints efficiently handle high stiffness, but when integrated in time they can suffer from instabilities in the nullspace directions, generating spurious transverse vibrations when pulling hard on thin inextensible objects or articulated rigid bodies. We show that geometric stiffness, the tensor encoding the change of force directions (as opposed to intensities) in response to a change of positions, is the missing piece between the two approaches. This previously neglected stiffness term is easy to implement and dramatically improves the stability of inextensible objects and articulated chains, without adding artificial bending forces. This allows time step increases up to several orders of magnitude using standard linear solvers.

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

Keywords: Physically based animation, Simulation, Dynamics, Constraints, Continuum mechanics, Geometric Stiffness

1 Introduction

Constraint-based simulation is very popular for implementing joints in articulated rigid bodies, and to enforce inextensibility in some directions of deformable objects such as cables or cloth. Its mathematical formulation makes it numerically robust to infinite stiffness, contrary to elasticity-based simulation, and some compliance can be introduced in the formulation or obtained through approximate solutions. Unfortunately, when the constraint forces are large, constraint-based objects are prone to instabilities in the transverse, unconstrained directions. This occurs when pulling hard on inextensible strings and sheets, or on chains of articulated bodies. The spurious vibrations can lead to unrealistic behaviors or even simulation divergence. They can be avoided using small time steps or complex non-linear solvers, however this dramatically slows down the simulation, while many applications, especially in interactive simulation, hardly allow for one linear solution per frame. The simulation speed can only be maintained by relaxing inextensibility, or using implicit elastic bending forces, however this changes the constitutive law of the simulated objects.

In this work, we show how to perform stable and efficient simulations of both extensible and inextensible constraint-based objects subject to high tensile forces. The key to transverse stability lies in the *geometric stiffness*, a first-order approximation of the change of direction of the internal forces due to rotation or bending. Neglecting the geometric stiffness, as usually done in constraint-based simulation, is a simplification of the linearized equation system, which in turn is a simplification of the exact, non-linear implicit integration. In case of thin objects, this leaves the transverse directions unconstrained, leading to uncontrolled extensions after time integration, introducing artificial potential energy. While this is acceptable for small stiffnesses or short time steps, this may introduce instabilities in the other cases. In this paper, we show that solving the complete linear equation allows high stiffnesses and large time steps which were only achievable using much slower non-linear solvers before. We show how to handle the geometric stiffness in a numerically stable way, even for very large material stiffness. The implementation is easy to combine with existing implicit solvers, and can provide several orders of magnitude speed-ups. Moreover, it allows a unification of rigid body and continuum mechanics.

In the next section, we detail our background and motivation through an introductory example. The principle of our method is then explained in Section 3. Its application to a wide variety of cases is then presented in Section 4. We conclude and sketch future work in Section 5.

2 Background and Motivation

The topic of constrained dynamics simulation is too wide to cite all the significant contributions. In this section, we briefly present the general approaches to enforce constraints: soft constraints (stiffness), hard and compliant constraints (Lagrange multipliers), position-based dynamics and others specialized methods. We refer the interested reader to [Nealen et al. 2006] for a more complete survey on the simulation of deformable objects and to [Witkin 1997; Erleben 2013] for constrained dynamics.

To help motivate our contributions and to review the previous work at the same time, we illustrate these approaches on a simple constraint: the distance between points.

2.1 Stiffness

A simple way to maintain desired distances between points is to use springs. Each spring applies a pair of opposite forces to its endpoints as a function of the distance (elasticity) and distance change rate (damping). In this comparison between methods, we neglect damping for the sake of simplicity. Let $\mathbf{x}_i$ and $\mathbf{x}_j$ be the positions of two particles, $\mathbf{v}_i$, $\mathbf{v}_j$ their velocities, and $\mathbf{a}_i$, $\mathbf{a}_j$ their accelerations. The potential energy of the spring is $V = k(\bar{l} - l)^2/2$ where $l = \|\mathbf{x}_i - \mathbf{x}_j\|$ is the current length of the spring, $\bar{l}$ the rest length, and k the stiffness. The intensity of the force is given by:

$$\lambda = -\frac{\partial V}{\partial l} = k(\bar{l} - l) \tag{1}$$

The 3D force is the product of the intensity with a unit vector parallel to the spring: $\mathbf{f}_i = -\mathbf{f}_j = \lambda \mathbf{u}$, where $\mathbf{u} = (\mathbf{x}_i - \mathbf{x}_j)/l$. Each particle is governed by Newton's Ordinary Differential Equation (ODE) $f = ma$, and they can be put together in the equation system: $\mathbf{M}\mathbf{a} = \mathbf{f}$, where $\mathbf{M}$ is a $n_d \times n_d$ diagonal matrix, n_d is the number of independent degrees of freedom (6 for two particles), while the $n_d \times 1$ state vectors $\mathbf{a}$ and $\mathbf{f}$ respectively are the net accelerations and the net forces. Discrete time integration is used to update positions and velocities. Using subscript $+$ to denote the value at the end of the incoming time step, the symplectic Euler method is given by:

$$\mathbf{v}_+ = \mathbf{v} + h\mathbf{a} \tag{2}$$

$$\mathbf{x}_+ = \mathbf{x} + h\mathbf{v}_+ \tag{3}$$

where h is the length of the time step. This method diverges for high stiffnesses or large time steps. A way to remedy this is implicit integration [Baraff and Witkin 1998]. The simplest version, very popular in Computer Graphics, is Implicit Euler, which replaces the current with the future accelerations in Eq.(2). This requires the solution of the linearized equation:

$$(\mathbf{M} - h^2\mathbf{K})\mathbf{v}_+ = \mathbf{p} + h\mathbf{f} \tag{4}$$

where $\mathbf{p} = \mathbf{M}\mathbf{v}$ is the momentum and $\mathbf{K} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ is the stiffness matrix. If needed, a symmetric damping term can be added to the system matrix [Baraff and Witkin 1998]. For clarity, we consider it part of the mass matrix in the rest of the paper.

Unfortunately, while typically sparse, the implicit integration matrix $\mathbf{H} = \mathbf{M} - h^2\mathbf{K}$ generally cannot be efficiently factored as a product of sparse matrices, and solving the linear system in Eq.(4) can be the bottleneck of the simulation, unless approximate iterative solutions are used. Moreover, since the elastic forces are invariant under rigid displacements, $\mathbf{K}$ is typically singular. Thus, large stiffnesses generally result in nearly singular matrices $\mathbf{H}$, which can strongly penalize the convergence speed of iterative methods. Consequently, simulating very stiff objects remains difficult using traditional elasticity.

2.2 Hard Constraints

Maintaining constant distances between particles to represent rigid links can be formulated as a Differential Algebraic Equation (DAE) using constraints [Barzel and Barr 1988; Ascher et al. 1995; Baraff 1996]. In this approach, besides the standard ODE $\mathbf{f} = \mathbf{M}\mathbf{a}$, each link is modeled using a holonomic constraint $\phi(\mathbf{x}) = 0$. There is a wide variety of constraint-based formulations depending on time discretization and constraint stabilization, and we simply sketch representative approaches. Combining the constraints with the ODE requires one or two time differentiations, depending on the time discretization used in the simulation:

$$\mathbf{J}\mathbf{v} = 0 \tag{5}$$

$$\mathbf{J}\mathbf{a} = \mathbf{c}(\mathbf{v}) \tag{6}$$

where the $n_c \times n_d$ matrix $\mathbf{J} = \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}}$ is the Jacobian of the constraints, n_c is the number of scalar constraints (one per link), and $\mathbf{c} = -\mathbf{J}\mathbf{v}$ is a velocity-dependent term encoding the nonlinearity of $\boldsymbol{\phi}$ with respect to $\mathbf{x}$. Each link is associated with one row of the Jacobian, composed of two non-null blocks $J_i = \mathbf{u}^T$ and $J_j = -\mathbf{u}^T$. Enforcing geometric constraints requires constraint forces. One can show that to preserve kinetic energy, the constraint forces $\mathbf{f}_c$ applied to the particles have to be in the form:

$$\mathbf{f}_c = \mathbf{J}^T\boldsymbol{\lambda} \tag{7}$$

The $\boldsymbol{\lambda}$ are called the Lagrange multipliers of the constraints. Putting it all together, implicit integration is implemented as a Karush-Kuhn-Tucker (KKT) equation system, here on velocities:

$$\begin{pmatrix} \mathbf{M} & -\mathbf{J}^T \\ \mathbf{J} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{v}_+ \\ \boldsymbol{\mu}_+ \end{pmatrix} = \begin{pmatrix} \mathbf{p} + h\mathbf{f}_e \\ 0 \end{pmatrix} \tag{8}$$

where $\boldsymbol{\mu} = h\boldsymbol{\lambda}$ are the constraint impulses, and $\mathbf{f}_e$ corresponds to the net external forces. Since $\mathbf{M}$ is easy to invert, one can come up with a smaller, though possibly denser, equation system using the Schur complement of the previous matrix:

$$\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\boldsymbol{\mu}_+ = -\mathbf{J}\mathbf{M}^{-1}(\mathbf{p} + h\mathbf{f}_e) \tag{9}$$

The right-hand term represents the time derivative of the constrained value (the velocity error) that would occur if no constraint forces were applied. The unknowns $\boldsymbol{\mu}$ represent the constraint impulse intensities necessary to cancel this error. The equation matrix in Eq.(9) is positive semi-definite, since it is singular in case of redundant constraints. When Eq.(9) is solved using a block Gauss-Seidel solver, the method is usually called *impulse-based* [Mirtich 1996; Weinstein et al. 2006]. This approach is easy to implement, straighforwardly handles constraint redundancy, and can be fast.

Hard constraints can be used to model thin deformable objects one can bend in the transverse (unconstrained) direction, while being theoretically inextensible in the constrained direction. However, when the tensile force is large with respect to the particle masses, instabilities in the transverse direction(s) occur, making the method impractical for such applications.

When an elastic object simulated using Eq.(4) undergoes kinematic constraints with Jacobian $\mathbf{J}$, the following equation holds:

$$\begin{pmatrix} \mathbf{H} & -\mathbf{J}^T \\ \mathbf{J} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{v}_+ \\ \boldsymbol{\mu}_+ \end{pmatrix} = \begin{pmatrix} \mathbf{p} + h\mathbf{f} \\ \boldsymbol{\psi} \end{pmatrix} \quad (10)$$

where the top row corresponds to Eq.(4) with additional constraint forces. A non-zero constraint value $\boldsymbol{\psi}$ may also account for non-holonomic constraints, or constraint stabilization [Cline and Pai 2003]. This formulation has been used for contacts between deformable objects [Duriez et al. 2008; Zheng and James 2011]. The reduced equation, not easily available since the upper-left block of the matrix is not efficiently invertible, has been addressed iteratively by [Otaduy et al. 2009]. [Kaufman et al. 2014] notice that the impulse-based or linear processing of contact forces in the transverse direction generate instabilities, and propose a third-degree model to avoid these. [Goldenthal et al. 2007] also identify such instabilities in the case of cloth simulation and propose an approximated, non-linear solver for handling them. We also address the issue of instabilities in the transverse directions, but our approach is first-degree and requires only one linear solution, using the stiffness of the constraint forces in contrast with the other methods.

2.3 Compliant Constraints

It is possible to add some relaxation to hard constraints by inserting a positive diagonal block in the bottom right of the matrix in Eq.(8). While sometimes presented as a numerical regularization for Eq.(9), this approach can be rigorously derived from an elastic constitutive law of the constraint forces: $\mathbf{C}\boldsymbol{\lambda} = -\boldsymbol{\phi}$, where the diagonal matrix $\mathbf{C}$ contains the compliance (inverse stiffness) of each scalar constraint [Servin et al. 2006; Lacoursière 2007]. Backwards time differentiation provides:

$$\mathbf{C}\boldsymbol{\lambda}_+ = \mathbf{C}\frac{\boldsymbol{\mu}_+}{h} = -\boldsymbol{\phi}_+ \approx -\boldsymbol{\phi} - h\mathbf{J}\mathbf{v}_+ \quad (11)$$

The equation system of Eq.(8) becomes:

$$\begin{pmatrix} \mathbf{M} & -\mathbf{J}^T \\ \mathbf{J} & \frac{1}{h^2}\mathbf{C} \end{pmatrix} \begin{pmatrix} \mathbf{v}_+ \\ \boldsymbol{\mu}_+ \end{pmatrix} = \begin{pmatrix} \mathbf{p} + h\mathbf{f}_e \\ -\frac{1}{h}\boldsymbol{\phi} \end{pmatrix} \quad (12)$$

with an analogous reduced equation given as

$$\left(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T + \frac{1}{h^2}\mathbf{C}\right)\boldsymbol{\mu}_+ = -\frac{1}{h}\boldsymbol{\phi} - \mathbf{J}\mathbf{M}^{-1}\left(\mathbf{p} + h\mathbf{f}_e\right). \quad (13)$$

This approach provides an appealing, unified treatment of elasticity and constraints. Notice the contrast with the implicit integration of stiff springs: while increasing stiffnesses leads to an increasingly high-valued, nearly singular matrix in Eq.(4), the corresponding decrease of the compliance in Eq.(13) keeps the equation numerically tractable. Consequently, high stiffnesses are more easily handled using the compliant constraint-based approach than the implicit integration of stiff elasticity. Moreover, matrix $\mathbf{C}$ has a numerical regularization effect which makes the constraint-based approach tractable even in degenerate cases.

Unfortunately, for stiff strings and surfaces, we observe the same instabilities as using hard constraints when pulling too hard, or lifting a heavy object using a light, inextensible string. This is due to the lack of stiffness in the transverse direction, resulting from the linearization of the constraints, as observed by [Kaufman et al. 2014]. It is possible to increase stability by solving the corresponding DAE using a non-linear solver, however this is compute-intensive and notoriously hard to implement robustly. Implicitly integrated bending stiffness can be used for stabilization [Servin and Lacoursière 2008], however it introduces ad-hoc parameters, additional complexity and changes the constitutive law. In summary, constraints efficiently model very stiff objects, as long as the tension is not too large, which defeats the purpose of stiffness.

2.4 Position-based Simulation

The trend of treating constraints and elasticity in an unified way can also be observed for position-based simulation [Gascuel and Gascuel 1994; Müller et al. 2006; Bender et al. 2013; Macklin et al. 2014], which generalizes impulse-based simulation to the position level. The particles first move unconstrained, generating constraint violations, then displacements are applied to cancel these, and velocities are updated accordingly. The iterative solution is implemented on a per-link basis, and easily generalized to all types of constraints. These approaches use block Gauss-Seidel-like solvers, which process the constraints sequentially and modify the positions after each constraint processing. The next constraint is thus processed using up-to-date positions, which efficiently handles non-linearities. Interestingly, no linearization artifact makes the correction overshoot, thus it does not suffer from the instabilities of previous constraint-based approaches. However, stiffness is not explicitly modeled in this method. The solution converges to a rigid behavior, and flexibility, obtained through approximate solutions, is a function of space and time discretization among other factors. Moreover, the convergence toward inextensibility can be very slow. Thus, while simple and stable, practical position-based simulation does not accurately handle stiff objects.

[Bouaziz et al. 2014] proposed a Jacobi-style solver with a similar way of updating constraints at each iteration. It results in a very efficient and stable method for specific cases when the matrix of Eq.(4) can be precomputed (constant mass matrix, constant material, constant set of constraints). This work also introduces continuum-based materials to position-based dynamics, however it only handles simplified physics (linear compressible materials) and moderate stiffness.

2.5 Specialized Approaches

In summary, simulating very stiff objects or constraints remains challenging. The implicit integration of stiff elastic forces is numerically hard, and requires complex non-linear solvers to achieve the desired stiffness. Hard constraints, or their extension to compliant materials, are numerically easier but instabilities occur in the unconstrained directions due to non-linearity. Position-based approaches handle non-linearity thanks to iterative local solutions which unfortunately are not efficient for stiff objects. Non-linear solvers have been proposed, but they require several linear solves per iteration and they do not accurately handle elasticity.

Specialized approaches have thus been proposed. The simplest way to enforce hard distance constraints is to use relative coordinates [Featherstone 1987; Bertails et al. 2006], however this is not well adapted to closed loops and it is not more stable. Specialized approaches have been proposed to simulate inextensible cables [García-Fernández et al. 2008; Servin et al. 2011; Sueda et al. 2011] and clothes [Provot 1995; Thomaszewski et al. 2009; Wang et al. 2010; Narain et al. 2012].

3 Stable Constraints

We present a new approach to take transverse directions into account while only solving a regular linear equation system. This formulation allows stable and fast simulation of non-linear energy-based constraints, such as the ones issued from the continuum mechanics. We first introduce concept of geometric stiffness using a simple example, then we show how we can use it to stabilize any constraint derived from an energy such as potentials and elastic constitutive laws.

3.1 Geometric Stiffness

The term *geometric stiffness* is generally employed in buckling analysis when linear approximations cannot handle large displacements for static analysis. It is associated with the geometrically non-linear part of the strain calculation. It is sometimes called *stability coefficient matrix* or *initial stress stiffness matrix*, see [Cook 1995; Zienkiewicz and Taylor 2000] for more details.

Coming back to our introductory example on springs, the stiffness matrix discussed in Sec.(2.1) is composed of four 3×3 matrix blocks computed as the sum of two terms:

$$\frac{\partial \mathbf{f}_j}{\partial \mathbf{x}_i} = \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_j} = -\frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_i} = -\frac{\partial \mathbf{f}_j}{\partial \mathbf{x}_j} = k\mathbf{u}\mathbf{u}^T - \frac{\lambda}{l}\left(\mathbf{I_3} - \mathbf{u}\mathbf{u}^T\right) \quad (14)$$

where $\mathbf{I_3}$ is the 3×3 identity matrix. The first term encodes the change of force corresponding to a change of spring length, thus a change of elastic energy and force intensity. The second term of the stiffness block encodes the change of force corresponding to a change of direction, as illustrated in Fig.(2). Note that the infinitesimal displacement orthogonal to the direction of the spring keeps the length and the elastic energy constant. The magnitude of the force does not change, only its direction does.

Based on Eq.(11), the stiffness of the compliant constraint is only $\mathbf{K} = \mathbf{J}^T\mathbf{C}^{-1}\mathbf{J}$, which corresponds to the extension term $k\mathbf{u}\mathbf{u}^T$ and neglects the geometric stiffness of the spring (which should be zero only for null extensions $l = \bar{l}$). One can easily check, by modifying an implicit spring simulation method, that neglecting the geometric stiffness causes the aforementioned instabilities in the transverse directions. In Projective Dynamics [Bouaziz et al. 2014], only the isotropic and constant term $-k\mathbf{I_3}$ is kept in Eq.(14) to allow pre-factorization, the full stiffness matrix being a Laplacian matrix. While this is accurate for $\bar{l} = 0$ or $l \gg \bar{l}$, the stiffness in the transverse direction is overestimated for small extensions $l \approx \bar{l}$, resulting in damping or locking.

3.2 Stiffness Splitting

We now present how to include the geometric stiffness term in a compliant constraint approach in order to get stable and more accurate elastic behaviors for arbitrary stiffnesses. While this can be derived from the variational formulation of implicit Euler and one step of Newton's method, our derivation extends [Servin et al. 2006; Lacoursière 2007], to emphasize the equivalence between hard constraints and the limit of infinitely stiff materials. Elastic forces act to minimize a certain potential energy:

$$V = \frac{1}{2}\boldsymbol{\phi}(\mathbf{x})^T\mathbf{C}^{-1}\boldsymbol{\phi}(\mathbf{x}) \quad (15)$$

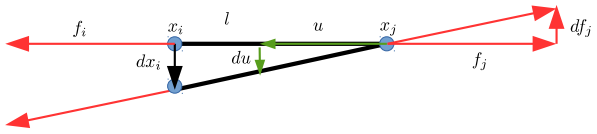


Figure 2: *Geometric stiffness. Transverse displacements change the direction of the force rather than its intensity.*

where $\boldsymbol{\phi}$ is a measure of departure from the rest configuration (*e.g.*, $(l - \bar{l})$ for a spring), $\mathbf{x}$ are the independent kinematic degrees of freedom, $\mathbf{C}^{-1}$ is the stiffness (*e.g.*, k for a spring). The elastic forces are computed as the opposite gradient of this potential energy:

$$\mathbf{f}_p = -\frac{\partial V}{\partial \mathbf{x}}^T = -\frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}}^T \frac{\partial V}{\partial \boldsymbol{\phi}}^T = \mathbf{J}^T\boldsymbol{\lambda} \quad (16)$$

where $\mathbf{J} = \frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}}$ is the constraint Jacobian (*e.g.*, $\pm\mathbf{u}^T$ for a spring), and $\boldsymbol{\lambda}$ is the constraint forces. The stiffness matrix is computed by differentiating the forces:

$$\mathbf{K} = \frac{\partial \mathbf{f}_p}{\partial \mathbf{x}} = \mathbf{J}^T\frac{\partial \boldsymbol{\lambda}}{\partial \boldsymbol{\phi}}\mathbf{J} + \frac{\partial \mathbf{J}^T}{\partial \mathbf{x}}\boldsymbol{\lambda} \quad (17)$$

where the first term is the *material stiffness* matrix representing the change of constraint force magnitude (*e.g.*, $\pm k\mathbf{u}\mathbf{u}^T$ for a spring), and the second term:

$$\widetilde{\mathbf{K}} = \frac{\partial \mathbf{J}^T}{\partial \mathbf{x}}\boldsymbol{\lambda} \quad (18)$$

is the *geometric stiffness* matrix encoding the variation of constraint force direction (*e.g.*, $\pm\frac{\lambda}{l}(\mathbf{I_3} - \mathbf{u}\mathbf{u}^T)$ for a spring).

These terms are used in the regular implicit integration system of Eq.(4). In this equation, the net force is the sum of external and elastic forces: $\mathbf{f} = \mathbf{f}_e + \mathbf{J}^T\mathbf{C}^{-1}\boldsymbol{\phi}$. In non-linear ODEs integrated semi-implicitly, $\boldsymbol{\phi}$ may be large, leading to inconsistent forces, artificial damping or even locking when $\mathbf{C} \to \mathbf{0}$.

The originality of our approach is to compute the elastic forces using the available constraint forces $\boldsymbol{\lambda}$ computed at the previous integration step, rather than the current $\boldsymbol{\phi}$. This is exactly equivalent for linear ODEs integrated using backward Euler, and this preserves force consistency in the case of non-linear ODEs. The implicit integration Eq.(4) becomes:

$$(\mathbf{M} - h^2(\mathbf{J}^T\mathbf{C}^{-1}\mathbf{J} + \widetilde{\mathbf{K}}))\mathbf{v}_+ = \mathbf{p} + h\mathbf{f}_e + h\mathbf{J}^T\boldsymbol{\lambda} \quad (19)$$

Noticing from Eq.(11) that $\boldsymbol{\lambda}_+ = \boldsymbol{\lambda} + h\mathbf{C}^{-1}\mathbf{J}\mathbf{v}_+$, we get the following dynamics equation:

$$(\mathbf{M} - h^2\widetilde{\mathbf{K}})\mathbf{v}_+ - h\mathbf{J}^T\boldsymbol{\lambda}_+ = \mathbf{p} + h\mathbf{f}_e \quad (20)$$

Using compliant constraint impulses and the constitutive law the same way as in Eq.(12), we get:

$$\begin{pmatrix} \mathbf{M} - h^2\widetilde{\mathbf{K}} & -\mathbf{J}^T \\ \mathbf{J} & \frac{1}{h^2}\mathbf{C} \end{pmatrix} \begin{pmatrix} \mathbf{v}_+ \\ \boldsymbol{\mu}_+ \end{pmatrix} = \begin{pmatrix} \mathbf{p} + h\mathbf{f}_e \\ -\frac{1}{h}\boldsymbol{\phi} \end{pmatrix} \quad (21)$$

This novel constrained dynamics equation is a middle ground between regular implicit elasticity and the constraint-based approach: the material stiffness is modeled using the compliance in the bottom right block, as in the compliant constraints (Eq.(12)), while the geometric stiffness is added to the mass matrix as usual in implicit integration (Eq.(4)). At the first time step, or if a constraint was not existing at the previous time step, the corresponding $\boldsymbol{\lambda}$ is unknown and we simply neglect its associated geometric stiffness. We have not noticed instabilities occurring in one time step.

To the best of our knowledge, this is the first time the geometric non-linearity of the constraints is handled like elasticity. Large, or even infinite stiffnesses are robustly handled thanks to the constraint-based formulation, while transverse instabilities are avoided thanks to the geometric stiffness. Hard kinematic constraints of Sec.(2.2) simply correspond to a null compliance matrix, but they are stabilized by the geometric stiffness contrary to traditional constrained dynamics. Similarly to [Teran et al. 2005], it is possible to enforce positive definiteness of the geometric stiffness matrix in order to improve numerical stability on certain solver classes.

Constraint-based elasticity unifies elasticity and constraint-based simulation, and leaves a modeling choice to the simulation designer. Large material stiffness is better handled using Lagrange multipliers with a compliance matrix in the bottom-right matrix block, however this creates a larger equation system. We call this the *compliance formulation*. Low stiffness is thus better handled in the top-left block using a stiffness matrix as in regular elasticity, which we call the *stiffness formulation*. Although our approach combines Lagrange multipliers and penalty forces, it should not be confused with the Augmented Lagrangian Method [Nocedal and Wright 2006] which requires several linear solves and only handles hard constraints in its standard formulation.

3.3 Unilateral Constraints

Unilateral constraints $\phi(\mathbf{x}) \geq 0$ are widely used to represent contacts and joint limits. For instance, the penetration depth at a contact point can be a non linear function of the independent DOFs that can benefit from our formulation to achieve better stability in case of large penetrations.

The extension of our approach to such constraints is straightforward. The geometric stiffness can be set in the (top-left block) dynamic matrix in the exact same way as for bilateral constraints, while the Signorini conditions are applied to the constraint velocities and impulses. In particular, in the case of springs the transverse motion will still be penalized *bilaterally*, while the extension motion will undergo unilateral constraints. The resulting KKT system can be readily solved using a general QP solver. When the dynamics matrix is easily invertible it is possible to compute the Schur complement in order to obtain an equivalent, but smaller Linear Complementarity Problem (LCP) that can be solved by various algorithms [Erleben 2013; Tomcin et al. 2014]. Coulomb friction can also be solved using constraint-based methods [Anitescu and Hart 2004; Kaufman et al. 2008]. Our method is compatible with all these numerical solvers.

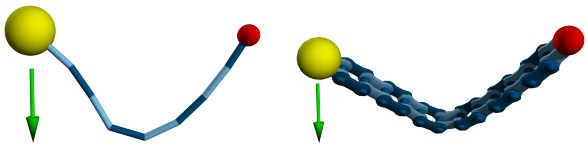


Figure 3: *Stress test for cables made of particles and distance constraints (left), or articulated rigid bodies (right). A heavy weight is falling down, eventually creating a large stress.*

4 Results

Potentially all non-linear constraints can be stabilized using geometric stiffness. We present here some examples to demonstrate the robustness of our method in various challenging situations. We stress that all the results presented in this section were obtained using a single linear system solve per time step, without extra constraint stabilization passes, using one thread of an Intel(R) Xeon(R) CPU @ 2.40GHz. Our solvers are available in the *Compliant* plugin of the SOFA open-source library [Faure et al. 2012].

4.1 Discrete Materials

Cables Significant improvements are achieved on the springs (*i.e.* distance constraints), most likely because they have two transverse directions for only one material direction. To test the distance constraints with our solver, we model an inextensible cable with null bending stiffness and mass properties close to that found on a crane (500Kg for 10m). We attach one extremity and we load the other with a heavy object, creating large mass ratios between the two. Starting the simulation with a tight cable, our constraints are able to pull very large weights with negligible extension using an LDLT solver. For a fixed 0.01s time step, without geometric stiffness, the simulation quickly diverges as shown in Table 1.

	weight	10t	100t	10^6t	10^9t	10^{12}t	10^{15}t
without	max err	$5e^{-14}$	X	X	X	X	X
with	max err	$5e^{-14}$	$1e^{-13}$	$9e^{-9}$	$7e^{-9}$	$8e^{-6}$	0.009

Table 1: *Error in percents of cable elongation to lift a box with different masses, with and without geometric stiffness for a fixed 0.01s time step.*

In the more challenging situation shown in Fig.(3), our solver is able to realistically handle $1:100$ mass ratios with large time steps (0.04s), while traditional constraints are not able to stably simulate the cable even for a $1:1$ mass ratio. Ten and one hundred distance constraints are respectively simulated at 500 fps and 100 fps using an LDLT solver. The cable swings naturally, with a slight damping due to integration.

Sheets Inextensible cloth simulation is challenging and has been mostly simulated using local strain limiting methods [Thomaszewski et al. 2009; Wang et al. 2010] that converge very slowly in large regions, or dedicated non-linear solvers [Goldenthal et al. 2007]. Using our method, inextensible or very stiff clothes can be simulated using a simple linear solution (50 MINRES iterations in the example shown in Fig.(4)), making it compatible with real-time applications. As a stress test, heavy weights were attached at the bottom of an inextensible cloth ($c = 0$). Using a direct LDLT solver the simulation is stable even with a very large mass ratio ($1:500$) between

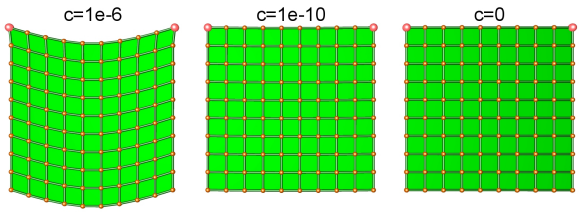


Figure 4: *Stiff clothes made of stable distance constraints*

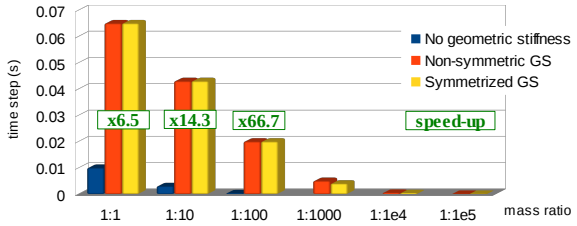


Figure 5: *Feasable constant time step for a varying weight attached to a chain of ten articulated rigids (ball-and-socket).*

the weights and the cloth particles and runs at more than 500 FPS.

Articulated Rigid Bodies can also be greatly stabilized using geometric stiffness. Consider a ball-and-socket joint connecting two rigid bodies with rigid coordinates $\mathbf{x}_1 = (\mathbf{t}_1, \mathbf{R}_1)$ and $\mathbf{x}_2 = (\mathbf{t}_2, \mathbf{R}_2)$ at points $\mathbf{s}_i = \mathbf{t}_i + \mathbf{R}_i \bar{\mathbf{s}}_i$, where the bar denotes local coordinates. The total Jacobian of the distance is: $\mathbf{J} = \begin{pmatrix} \mathbf{J}_1 & -\mathbf{J}_2 \end{pmatrix}$ where $\mathbf{J}_i = \begin{pmatrix} \mathbf{I_3} & -\widehat{\mathbf{R}_i \bar{\mathbf{s}}_i} \end{pmatrix}$, and a hat on a vector represents the 3×3 skew-symmetric matrix encoding the associated vector product. The geometric stiffness associated with a constraint force $\boldsymbol{\lambda}$ is:

$$\widetilde{\mathbf{K}} = \frac{\partial \mathbf{J}^T}{\partial \mathbf{x}} \boldsymbol{\lambda} = \begin{pmatrix} \widetilde{\mathbf{K}}_1 & 0 \\ 0 & -\widetilde{\mathbf{K}}_2 \end{pmatrix}, \quad \widetilde{\mathbf{K}}_i = \begin{pmatrix} 0 & 0 \\ 0 & \widehat{\boldsymbol{\lambda} \mathbf{R}_i \bar{\mathbf{s}}_i} \end{pmatrix} \tag{22}$$

Note that the geometric stiffness matrix is not symmetric in this case. This is due to the fact that rigid transformations are not a vector space. While non-symmetric linear systems are relatively easy to solve (using a LU decomposition for instance), non-symmetric LCPs are more difficult to handle as most solvers expect at least a symmetric matrix. To cope with this issue, we use the symmetric part of the geometric stiffness: $(\widetilde{\mathbf{K}} + \widetilde{\mathbf{K}}^T)/2$. We found this approach to have virtually the same stabilizing properties as using the exact, non-symmetric geometric stiffness matrix, as shown in Fig. 5.

As an added benefit, in most cases our approach removes the need for a separate constraint stabilization pass, traditionally required in rigid-body simulations. We can instead specify a very small, non-zero compliance value and still obtain a stable behavior with no constraint drift. This contrasts with Baumgarte stabilizers which are difficult to tune. For larger time steps, an extra stabilization pass (*e.g.*, post-stabilization [Cline and Pai 2003]) can be applied. For the stress-test shown in Fig.(3) of a chain of rigid bodies connected by ball-and-socket joints, we observe significant improvements reported in Fig.(5). Compared to standard constraint solving approaches, our simulator can handle much larger mass ratios, up to 3 orders of magnitude. This higher stability can be exploited to increase the time step by more than one order of magnitude for regular simulations (mass ratio $\leq$1:10), but more challenging situations can also be achieved with real-time compatible time steps. As a comparison, both commercial simulators PhysX and Havok recommend a mass ratio smaller than $1:10$.

While simulating a long chain (*e.g.* 50 rigid bodies) is very hard using regular constraints, it is possible with our solver. The accompanying video shows the same example with hinge joints. Without geometric stiffness, the chain keeps vibrating even using a mass ratio of $1:1$ and a tiny time step. Using the geometric stiffness, the same chain is very stable even for large time steps. It is possible to create very long chains and to add joint limits similar to the ones found in bicycles.

4.2 Continuous Materials

Interestingly, the compliant formulation of Eq.(21) can be applied to continuous media mechanics methods such as FEM, to combine accuracy and stability. In continuum mechanics, the elastic potential energy of deformable materials is typically integrated in space using Gaussian cubature, *i.e.*, as a weighted sum of energy densities computed at sample locations in the material:

$$V = \sum_i \frac{1}{2} w_i \boldsymbol{\varepsilon}_i^T \mathbf{D}_i \boldsymbol{\varepsilon}_i \tag{23}$$

where, for each Gauss point i, $\boldsymbol{\varepsilon}_i$ is the strain (vector written using Voigt notation), $\mathbf{D}_i = \partial \boldsymbol{\sigma}_i / \partial \boldsymbol{\varepsilon}_i$ is the material stiffness matrix characterizing the local elastic behavior, $\boldsymbol{\sigma}_i$ is the stress, and w_i is the weight, a fraction of the volume of the integration domain.

It is natural to express constraints at the material strain level (*i.e.*, $\boldsymbol{\phi} = \boldsymbol{\varepsilon}$), as it results in a block-diagonal compliance matrix, with one constraint $\mathbf{D}_i^{-1}$ per Gauss point i. Stiffness blocks are small (resp. 1×1, 3×3, 6×6 for 1-, 2-, 3-dimensional materials with a regular strain measure) and can be independently and easily inverted. For linear materials, they are constant and can be precomputed at initialization. Sometimes the compliance formulation $\mathbf{D}_i^{-1}$ is even known analytically (*e.g.*, (an)isotropic Hooke's law).

Strain is usually derived from the deformation gradient $\mathbf{F}$, that is linear with respect to $\mathbf{x}$ for most deformation models (finite elements, mesh-less methods, linear blend skinning, modal subspace). Therefore, the geometric stiffness depends on the non-linearity of the strain measure $\boldsymbol{\varepsilon}(\mathbf{F})$. For instance, the Green-Lagrange case is derived in Appendix A. The co-rotational strain measure is: $\boldsymbol{\varepsilon} = (\mathbf{R}^T \mathbf{F} - \mathbf{I})^*$, where $\mathbf{R}$ is the closest rotation from $\mathbf{F}$ and $(.)^*$ maps a tensor to a vector in Voigt notation. The Jacobian $\mathbf{J} = \partial \boldsymbol{\varepsilon} / \partial \mathbf{x}$ is updated at each time step using the current estimate of $\mathbf{R}$; while the geometric stiffness is obtained from the derivative of $\mathbf{R}$ with respect to $\mathbf{F}$. For the SVD-based corotational strain and principal stretches [Irving et al. 2004] the geometric stiffness is based on the jacobian of the SVD [Papadopoulo and Lourakis 2000]. This method works similarly for any other strain measures (invariants of Green's deformation tensor, logarithmic strain, etc.), and for any material laws (hyperelasticity, anisotropy, plasticity). Any continuum-based energy can be simulated using constraints with any discretization method. Note that the geometric stiffness of the corotational strain measure is often neglected in the *stiffness* formulation, which can lead to instabilities for large deformations. The proper handling of the geometric stiffness makes the compliance formulation suitable for Continuous Media Mechanics. Its implementation is available in the *Flexible* plugin of SOFA [Faure et al. 2012].

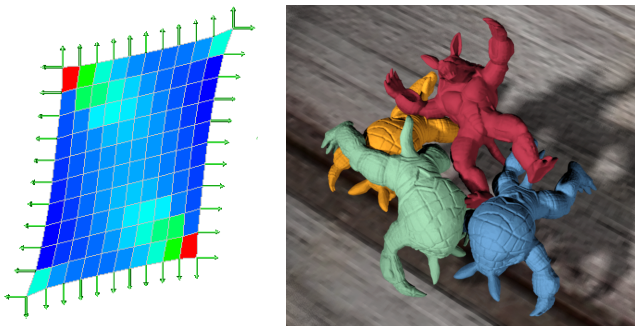


Figure 6: *Complex materials. Left: A patch of anisotropic quads pulled by external forces (Von Mises Stress texturing). Right: Four armadillos, all simulated by different materials (red rigid body, green plastic tetra FEM, blue Neo-Hookean hexa FEM, yellow St Venant-Kirchhoff meshless), are colliding using unilateral constraints.*

2D Materials Simulating cloth using hard constraint distances is challenging, but using stiff 2D continuum-material is even more so. We thus compare the standard implicit stiffness with our continuum-based constraints on stiff triangular finite elements (St Venant Kirchoff with $E = 1$GPa, $\nu = 0.49$). Using a MINRES solver, the stiffness version never converges (even after 10000 iterations), while our method converges in about 1000 iterations. It is fairly stable and several hundred Kg can even be attached at the bottom of the cloth without stretching it.

In Table 2 we analyze the conditioning of linear systems for cloth animation modeled with both distance constraints and triangular FEM. The maximal condition number is always much smaller with our approach (KKT system) than with implicit stiffness, which could explain why the numerical solver converges more easily.

1D Springs (Fig. 4)	compliance (c=0)	O(1e5)
	stiffness (k=1e15)	O(1e12)
2D St-VK triangle FEM ($E = 1$ GPa, $\nu = 0.49$)	compliance	O(1e3)
	stiffness	O(1e6)

Table 2: *Maximal condition number during the animation of a falling stiff cloth (10x10)*

3D Materials As already noticed, for soft enough materials the *stiffness* formulation is more suitable in terms of computational cost. However only soft material allow behavior comparisons with the *compliance* formulation. We thus compared the two formulations for a soft cantilever beam, leading to similar results as expected. This validates the correctness of our method. Likewise, neglecting the geometric stiffness leads to incorrect behaviors (*i.e.*, transverse oscillations). However, in 3D there are no pure unconstrained transverse directions as in thin objects. Consequently, simulating 3D materials takes no advantage of the constraint formulation, and does not allow to easily simulate higher stiffnesses. Moreover, the number of constraints necessary to simulate 3D elasticity becomes really large: a finite element beam composed of 525 nodes and 2560 tetrahedra requires 15360 scalar constraints (6 per integration point) to simulate its elasticity, compared with 1575 scalar DOFs in the stiffness approach. It becomes hard to redeem the system size by improving its conditioning.

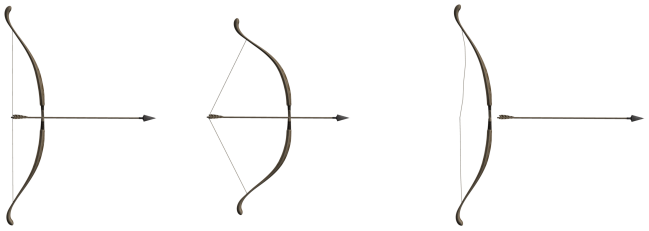


Figure 7: *A bow physically shooting an arrow. (Left) At rest, an inextensible string slightly bends a stiff solid wood frame. (Middle) Applying a force on the string significantly bends the frame. (Right) When released, the string is tightened by the wood which comes back to its rest position, projects the arrow then oscillates.*

Complex Materials An example of anisotropic material, is shown in Fig.6-Left. A patch of quads FEM with a Green-Lagrange strain measure and an orthotropic Hooke's law (matrix $\mathbf{D}$) is pulled by external forces. Note that a bi-linear quad has 4 integration points meaning 3×4 scalar constraints.

Fig. 6-Right presents others material laws, strain measures and types of DOFs. A non-linear hyperelastic material is demonstrated with a Neo-Hookean material computed from principal stretches. The compliance matrix needs to be updated at each time step. Principal stretches are interesting in the case of constraints as they create only 3 constraints in 3D (*vs* 6 for a regular strain). The material matrices inverted to get the compliance matrices are simple 3×3 matrices and the size of the compliance part of the system is twice as small. Plasticity can be simulated as a linear mapping from strain to strain which does not generate any geometric stiffness [Irving et al. 2004].

4.3 Complex Objects

Ragdoll The wild ragdoll presented in Fig.(1) is made of 12 rigid bodies articulated with joints limits (unilateral constraints). It is swinging on an articulated rigid liana composed of 10 ball-and-socket links. The simulation runs at 70 fps with a 0.01s time step using a Projected Gauss-Seidel LCP solver (on the Schur Complement computed from a LDLT decomposition) with 200 iterations. It swings naturally, and remains stable even when strongly shaken by user interaction. Conversely, neglecting the geometric stiffness produces an unstable simulation.

Bow The bow shown in Fig.7 is composed of a frame and a string. The frame is a 3D material simulated by a corotational meshless method with a stiff wood material ($E = 10$ GPa). The string is made of ten inextensible distance constraints. The string is attached to the frame using hard bindings [Sifakis et al. 2007]. The two-way interaction between the string and the arrow (rigid body) is modeled as an unilateral constraint. Accurately handling high stiffnesses is necessary to simulate the shooting. Our solver runs at 85 fps, $dt = 0.005$s with collisions (Projected Gauss-Seidel).

Trampoline We used nearly inextensible distance constraints ($c = 10^{-10}$) to model the jumping mat of a trampoline connected to a rigid frame by soft springs $k = 5.10^3$ as illustrated in Fig.(1). This example combines a stiffness formulation for the springs with a compliance formulation for the cloth, while the contacts are handled using hard unilateral constraints. The simulation runs at more than 35 fps with

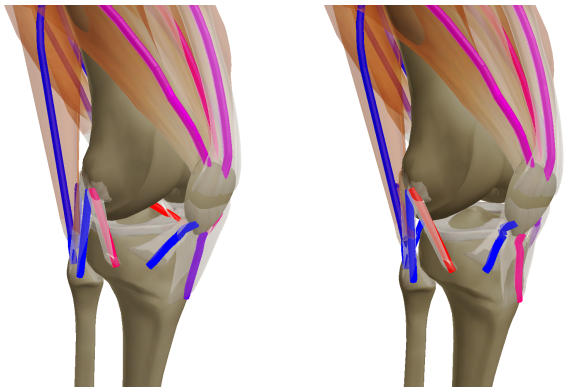


Figure 8: *Stable biomechanical simulation of the Lachman test using our 1D elastic models with stiff unilateral constraints. Left: normal translation when pulling the tibia forward. Right: excessive translation due to a deficient anterior cruciate ligament.*

a Projected Gauss-Seidel LCP solver (on the Schur Complement computed from a LDLT decomposition) with a 0.01s time step (contacts are the limitation for the time step). Using a stiffness formulation, we were not able to avoid the divergence of the jumping mat due to numerical hardness of the nearly inextensible material.

Knee Figure 8 shows the simulation of a knee with ligaments and muscle-tendon units using rigid bodies in contact, connected by curved springs. The spring particles as well as the muscle, tendons and ligament surfaces are attached to the bones using linear blend skinning. The independent degrees of freedom (DOFs) are the translations and rotations of the bones, and the spring forces are propagated to them using the transposed of the skinning Jacobian $\mathbf{J}_s$. Since ligaments and tendons becomes very stiff after an elongation of about 5%, we model these bi-phasic materials by combining two 1D linear elastic laws: a soft one formulated as a stiffness, and a stiff one above a strain limit (*compliance* formulation with unilateral constraints). Applying varying strain limits allows us to simulate hamstring muscles contractions to flex the knee. The simulation runs very responsively at 11 fps, using an active-set MINRES-based QP solver. The geometric stiffness $\mathbf{K}_g$ of the embedded springs is propagated to the DOFs as $\mathbf{J}_s^T \mathbf{K}_g \mathbf{J}_s$. Without it, the simulation is stable only for very small time steps, or with large damping ratios, resulting in less interactivity or instabilities as can be seen in the accompanying video.

5 Conclusion and Future Work

We have shown the importance of geometric stiffness for the dynamic simulation of stiff objects. This previously overlooked linear tensor only changes the stiffness matrix, and provides accurate first-order force approximations, with simple implementation which can be used in all elasticity-based and constraint-based implicit simulation methods.

Some limitations of our work are left for future work. Stiff damping is a straightforward extension. More non-linear constraints will be handled. The stability, while dramatically better than before, is not guaranteed since geometric stiffness only provides a more accurate first-order operator, not a real non-linear solution. In the compliant formulation, the mass-stiffness matrix looses its block-diagonal structure, which makes it more complex to factor. Numerical criteria for choosing between constraint and elasticity could be useful.

Compliant constraints with geometric stiffness allow the stable simulation of arbitrarily stiff behaviors and large mass ratios with larger time steps than previous methods. They are compatible with standard elasticity, and provide an alternative way to model constitutive laws. This is especially useful for large and infinite stiffnesses, which can be simulated using a single linear solution at each frame, while previous methods require complex non-linear solvers, smaller time steps or modified constitutive laws with ad-hoc parameters. This approach does not require pre-factorizations, and thus allows creating or deleting constraints on-the-fly, which is useful for artistic control. We have successfully applied it to mass-springs systems, 2D and 3D continuum-based elastic models as well as articulated rigid bodies with kinematic loops, with accelerations up to several orders of magnitude with respect to previous methods. We believe that its combined simplicity, efficiency and versatility, make it a useful base tool for mechanical simulation.

Acknowledgments

The authors wish to thank Laura Paiardini for her help with the models, renderings and the video editing, Romain Testylier for the corkscrew model and Midori Hyndman for the proofreading. We also thank the anonymous reviewers for the suggestions for improving the paper. This work was supported by funding from French FUIs *Dynamit* and *Collodi*, and French ANR project SoHusim. This work was partially funded by the advanced grant no. 291184 EXPRESSIVE from the European Research Council (ERC-2011-ADG_20110209). The implementation was performed using the SOFA open-source library (www.sofa-framework.org) supported by INRIA and partners.

References

ANITESCU, M., AND HART, G. D. 2004. A fixed-point iteration approach for multibody dynamics with contact and small friction. *Mathematical Programming 101*.

ASCHER, U. M., H.CHIN, L.R.PETZOLD, AND REICH, S. 1995. Stabilization of Constrained Mechanical Systems with DAEs and Invariant Manifold. *Journal of Mechanics of Structures and Machines 23*.

BARAFF, D., AND WITKIN, A. 1998. Large steps in cloth simulation. In *Proc. 25th annual conference on Computer graphics and interactive techniques (ACM SIGGRAPH)*.

BARAFF, D. 1996. Linear-time Dynamics Using Lagrange Multipliers. In *Proc. 23th annual conference on Computer graphics and interactive techniques (ACM SIGGRAPH)*.

BARZEL, R., AND BARR, A. H. 1988. A Modeling System Based On Dynamic Constraints. In *Computer Graphics (Proc. SIGGRAPH)*.

BENDER, J., MÜLLER, M., OTADUY, M. A., AND TESCHNER, M. 2013. Position-based Methods for the Simulation of Solid Objects in Computer Graphics. In *Computer Graphics Forum (Eurographics State of the Art Report)*.

BERTAILS, F., AUDOLY, B., CANI, M.-P., QUERLEUX, B., LEROY, F., AND LÉVÊQUE, J.-L. 2006. Super-helices for Pre-

dicting the Dynamics of Natural Hair. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*.

BOUAZIZ, S., MARTIN, S., LIU, T., KAVAN, L., AND PAULY, M. 2014. Projective Dynamics: Fusing Constraint Projections for Fast Simulation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*.

CLINE, M. B., AND PAI, D. K. 2003. Post-stabilization for rigid body simulation with contact and constraints. In *International Conference on Robotics and Automation*.

COOK, R. D. 1995. *Finite Element Modeling for Stress Analysis*. John Wiley & Sons, Inc.

DURIEZ, C., DUBOIS, F., KHEDDAR, A., AND ANDRIOT, C. 2008. Realistic Haptic Rendering of Interacting Deformable Objects in Virtual Environments. *CoRR*.

ERLEBEN, K. 2013. Numerical Methods for Linear Complementarity Problems in Physics-based Animation. In *ACM SIGGRAPH Courses*.

FAURE, F., DURIEZ, C., DELINGETTE, H., ALLARD, J., GILLES, B., MARCHESSEAU, S., TALBOT, H., COURTECUISSE, H., BOUSQUET, G., PETERLIK, I., AND COTIN, S. 2012. SOFA: A Multi-Model Framework for Interactive Physical Simulation. In *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*. Springer. `http://www.sofa-framework.org`.

FEATHERSTONE, R. 1987. *Robot Dynamics Algorithm*. Kluwer Academic Publishers, Norwell, MA, USA.

GARCÍA-FERNÁNDEZ, I., PLA-CASTELLS, M., AND MARTÍNEZ-DURÁ, R. J. 2008. Elevation Cable Modeling for Interactive Simulation of Cranes. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.

GASCUEL, J.-D., AND GASCUEL, M.-P. 1994. Displacement constraints for interactive modeling and animation of articulated structures. *Visual Computer 10*, 4.

GOLDENTHAL, R., HARMON, D., FATTAL, R., BERCOVIER, M., AND GRINSPUN, E. 2007. Efficient Simulation of Inextensible Cloth. *ACM Transactions on Graphics (Proc. SIGGRAPH)*.

IRVING, G., TERAN, J., AND FEDKIW, R. 2004. Invertible finite elements for robust simulation of large deformation. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.

KAUFMAN, D. M., SUEDA, S., JAMES, D. L., AND PAI, D. K. 2008. Staggered Projections for Frictional Contact in Multibody Systems. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*.

KAUFMAN, D. M., TAMSTORF, R., SMITH, B., AUBRY, J.-M., AND GRINSPUN, E. 2014. Adaptive Nonlinearity for Collisions in Complex Rod Assemblies. *ACM Transaction on Graphics (Proc. SIGGRAPH)*.

LACOURSIÈRE, C. 2007. *Ghosts and machines: regularized variational methods for interactive simulations of multibodies with dry frictional contacts*. PhD thesis, Umeå University.

MACKLIN, M., MÜLLER, M., CHENTANEZ, N., AND KIM, T.-Y. 2014. Unified Particle Physics for Real-Time Applications. *ACM Transactions on Graphics (Proc. SIGGRAPH)*.

MIRTICH, B. 1996. *Impulse-based Dynamic Simulation of Rigid Body Systems*. PhD thesis, University of California.

MÜLLER, M., HEIDELBERGER, B., HENNIX, M., AND RATCLIFF, J. 2006. Position Based Dynamics. *Proc. VRIPhys*.

NARAIN, R., SAMII, A., AND O'BRIEN, J. F. 2012. Adaptive Anisotropic Remeshing for Cloth Simulation. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*.

NEALEN, A., MÜLLER, M., KEISER, R., BOXERMAN, E., AND CARLSON, M. 2006. Physically Based Deformable Models in Computer Graphics. *Computer Graphics Forum (Eurographics State of the Art Report)*.

NOCEDAL, J., AND WRIGHT, S. J. 2006. *Numerical Optimization (2nd ed.)*. Springer-Verlag.

OTADUY, M. A., TAMSTORF, R., STEINEMANN, D., AND GROSS, M. 2009. Implicit Contact Handling for Deformable Objects. *Computer Graphics Forum (Proc. of Eurographics)*.

PAPADOPOULO, T., AND LOURAKIS, M. I. A. 2000. Estimating the jacobian of the singular value decomposition: Theory and applications. In *Proc. European Conference on Computer Vision*.

PROVOT, X. 1995. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *Proc. Graphics Interface*.

SERVIN, M., AND LACOURSIÈRE, C. 2008. Rigid Body Cable for Virtual Environments. *IEEE Transactions on Visualization and Computer Graphics 14*.

SERVIN, M., LACOURSIÈRE, C., AND MELIN, N. 2006. Interactive simulation of elastic deformable materials. In *Proc. SIGRAD*.

SERVIN, M., LACOURSIÈRE, C., NORDFELTH, F., AND BODIN, K. 2011. Hybrid, Multiresolution Wires with Massless Frictional Contacts. *IEEE Trans. on Visualization and Computer Graphics 17*.

SIFAKIS, E., SHINAR, T., IRVING, G., AND FEDKIW, R. 2007. Hybrid Simulation of Deformable Solids. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.

SUEDA, S., JONES, G. L., LEVIN, D. I. W., AND PAI, D. K. 2011. Large-scale Dynamic Simulation of Highly Constrained Strands. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*.

TERAN, J., SIFAKIS, E., IRVING, G., AND FEDKIW, R. 2005. Robust quasistatic finite elements and flesh simulation. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.

THOMASZEWSKI, B., PABST, S., AND STRASSER, W. 2009. Continuum-based Strain Limiting. *Computer Graphics Forum (Proc. Eurographics)*.

TOMCIN, R., SIBBING, D., AND KOBBELT, L. 2014. Efficient enforcement of hard articulation constraints in the presence of closed loops and contacts. *Computer Graphics Forum (Proc. Eurographics)*.

WANG, H., O'BRIEN, J. F., AND RAMAMOORTHI, R. 2010. Multi-Resolution Isotropic Strain Limiting. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*.

WEINSTEIN, R., TERAN, J., AND FEDKIW, R. 2006. Dynamic simulation of articulated rigid bodies with contact and collision. *IEEE Transactions on Visualization and Computer Graphics*.

WITKIN, A. 1997. Physically Based Modeling: Principles and Practice – Constrained Dynamics. *Computer Graphics*.

ZHENG, C., AND JAMES, D. L. 2011. Toward High-Quality Modal Contact Sound. *ACM Transactions on Graphics (Proc. SIGGRAPH)*.

ZIENKIEWICZ, O. C., AND TAYLOR, R. L. 2000. *The Finite Element Method: Solid Mechanics*, 5 ed., vol. 2. Butterworth-Heinemann.

A Geometric Stiffness for Green-Lagrangian Strain

We detail the computations for one Gauss point of a 3-dimensional material which deformations are measured using Green-Lagrangian strain. Computations for 1D and 2D materials and other strain measures follow the same methodology.

For a material point p, with undeformed position $\bar{p}$, the deformation gradient is the 3×3 matrix $\mathbf{F} = \partial p / \partial \bar{p}$. In most popular discretization methods (*e.g.,* finite elements, linear blend skinning, moving least squares, linear modal subspace), it is computed as a linear combination of the degrees of freedom $\mathbf{x}$ [Cook 1995; Zienkiewicz and Taylor 2000]. Noting the deformation gradient as a 9-dimensional row-major vector $\mathbf{F}^{\times}$, we have $\mathbf{F}^{\times} = \mathbf{B}\mathbf{x}$ where $\mathbf{B}$ is a $9 \times n_d$ constant matrix containing the values of the shape function gradients evaluated at $\bar{p}$. The 6-dimensional strain vector, computed using the Green-Lagrangian measure, is $\boldsymbol{\varepsilon} = \frac{1}{2}(\mathbf{F}^T\mathbf{F} - \mathbf{I_3})^*$ where the operator $()^*$ vectorizes a symmetric strain tensor using Voigt notation. By differentiation, we obtain the $6 \times n_d$ Jacobian matrix:

$$\mathbf{J} = \frac{\partial \boldsymbol{\varepsilon}}{\partial \mathbf{x}} = \frac{\partial \boldsymbol{\varepsilon}}{\partial \mathbf{F}^{\times}} \frac{\partial \mathbf{F}^{\times}}{\partial \mathbf{x}} = \begin{bmatrix} F_{11} & 0 & 0 & F_{21} & 0 & 0 & F_{31} & 0 & 0 \\ 0 & F_{12} & 0 & 0 & F_{22} & 0 & 0 & F_{32} & 0 \\ 0 & 0 & F_{13} & 0 & 0 & F_{23} & 0 & 0 & F_{33} \\ F_{12} & F_{11} & 0 & F_{22} & F_{21} & 0 & F_{32} & F_{31} & 0 \\ 0 & F_{13} & F_{12} & 0 & F_{23} & F_{22} & 0 & F_{33} & F_{32} \\ F_{13} & 0 & F_{11} & F_{23} & 0 & F_{21} & F_{33} & 0 & F_{31} \end{bmatrix} \mathbf{B} \tag{24}$$

Finally, we obtain the $n_d \times n_d$ geometric stiffness matrix through another differentiation:

$$\widetilde{\mathbf{K}} = \frac{\partial \mathbf{J}^T}{\partial \mathbf{x}} \boldsymbol{\lambda} = \mathbf{B}^T \begin{bmatrix} \boldsymbol{\lambda}^{\wedge} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\lambda}^{\wedge} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\lambda}^{\wedge} \end{bmatrix} \mathbf{B} \tag{25}$$

where the operator $()^{\wedge}$ converts a Voigt stress vector to a symmetric matrix.