Optimized Damping for Dynamic Simulations

Ruediger Schmedding * Marc Gissler† Matthias Teschner‡
University of Freiburg University of Freiburg University of Freiburg

Abstract

Dynamic simulations can benefit a lot from an appropriate damping approach. For example, the stability is improved and a larger time step can be chosen. Furthermore, badly shaped meshes, e. g. containing sharp angles or slivers, can be handled if a proper damping approach is used. However, it must be ensured that the damping forces do not change the global movement of the object, i. e. they have to preserve linear and angular momentum. In this paper, we present a novel damping approach that is based on iterative spring damping to further improve the stability. We show that the resulting forces can be computed directly without actually performing the iterations. The approach does not require any connectivity information about the object and therefore, it can be used for arbitrary object representations. Further, it is independent of the integration scheme and the deformation model. The approach provides a simple parameter setting and guarantees that the damping forces do not overshoot. Finally, we illustrate that our approach allows for larger time steps compared to existing damping methods.

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 and Virtual Reality

Keywords: Physically Based Animation, Deformable Modeling, Stability, Damping

1 Introduction

Damping is a very relevant topic in many areas of computer animation. On the one hand, damping can significantly influence the stability, i. e. the time step of a dynamic simulation and, thus, the perceived performance of an animation. On the other hand, damping also influences the dynamic motion of structures. The resulting effects can be either desired, e. g. reduced oscillations of a deformable body, or disturbing, e. g. an unnatural slow-down of a motion.

In general, a damping force term $\mathbf{C}\dot{\mathbf{X}}$ is introduced into the equation of motion

$$\mathbf{M}\ddot{\mathbf{X}} + \mathbf{C}\dot{\mathbf{X}} = \mathbf{F} - \mathbf{K}\mathbf{X} \tag{1}$$

with $\mathbf{M}$ denoting the mass matrix, $\mathbf{X}$ denoting the positions of all points of the object with its first and second derivative $\dot{\mathbf{X}}$ and $\ddot{\mathbf{X}}$, $\mathbf{K}$ denoting the stiffness matrix that represents the force-displacement-relation and $\mathbf{F}$ denoting the external forces. $\mathbf{C}$ is a user-defined matrix that introduces a velocity-dependent damping force. To decouple the system of differential equations, mass lumping is commonly assumed to get a diagonal mass matrix. Similarly, either the damping matrix has to be diagonal (see, for example, [Müller et al. 2002]), or the damping forces have to be computed in an additional step. The first alternative leads to the following equation of motion for a single mass point:

$$m\ddot{\mathbf{x}} + \gamma\dot{\mathbf{x}} = \mathbf{f}, \tag{2}$$

where m is the mass of the point, $\mathbf{x}$ its position, $\mathbf{f}$ the force acting on it, including the internal force resulting from $\mathbf{KX}$ in (1), and γ is a user-defined parameter. The second alternative leads to the equation

$$m\ddot{\mathbf{x}} = \mathbf{f} + \mathbf{f}^d, \tag{3}$$

where $\mathbf{f}^d$ denotes a damping force that not only depends on one, but on various mass points.

Eq. (2) is commonly known as point damping and only damps the absolute velocities of the points. In contrast, (3) allows a variety of possible computations of $\mathbf{f}^d$, including, for example, the damping of relative velocities. Note that $\mathbf{f}^d$ generally depends on the damping parameter γ.

The objective of the damping configuration is to find a setting that maximizes the stability, while undesired effects are avoided or within an acceptable range. For example, the point damping force $\gamma\dot{\mathbf{x}}$ from (2) results in an improved numerical stability, but at the price of preventing the acceleration of a mass point. This behavior is desired in the specific context of friction and might be appropriate in some configurations, but in general, the effect is undesired in an animation.

Therefore, damping terms commonly employ relative velocities for the computation of $\mathbf{f}^d$ in (3) to avoid the undesired influence on the global movement. For example, the relative velocity of adjacent mass points can be used. In this case, damping forces are computed for pairs of points and are applied symmetrically to both points. This method preserves the linear momentum of the point pair and, thus, of the entire structure. For the conservation of angular momentum, a projection technique has to be applied. Then, this type of damping only influences relative movements of points, i. e. internal oscillations, while the global movement of the structure is not affected. However, estimating suitable damping parameters is still a challenging task.

1.1 Contribution

In this paper, we present a new relative damping approach that is based on iterative spring damping. While the iterative computation of spring damping forces improves the stability of dynamic simulations, it is less efficient compared to the original spring damping (see e. g. [Nealen et al. 2006]). Therefore, we also present an efficient direct computation of the damping forces that result from the iterative process. These forces automatically preserve the linear momentum. In order to preserve the angular momentum, we

*e-mail: schmedd@informatik.uni-freiburg.de
†e-mail:gissler@informatik.uni-freiburg.de
‡e-mail:teschner@informatik.uni-freiburg.de

introduce two projection schemes and one Linear Programming approach that eliminate the torque. In contrast to existing methods, our approach does not need any connectivity information of the object and therefore, it can be used for arbitrary object representations. Further, our approach can be implemented within any constitutional model, even meshless approaches, and is independent of the employed integration scheme. It can be used as one-dimensional damping on edges, two-dimensional on surfaces or cloth and also for damping of volumetric elements. The damping constant γ is always within the range between 0 and 1 and thus, it simplifies the parameter setting. Finally, larger time steps can be chosen using our approach.

1.2 Previous work

Deformable modeling in Computer Graphics started with the pioneering work of Terzopoulos [1987]. Since then, a great variety of models for simulating rigid bodies, deformable objects, fluids and gases has been developed. Among these are various mass-spring-approaches [Chen et al. 1998; Bourguignon and Cani 2000], potential based approaches [Teschner et al. 2004], geometrically motivated methods [Müller et al. 2005] and Finite Element simulations [Hauth and Strasser 2004; Müller and Gross 2004]. In most models, some kind of damping is included. We refer the reader to the report by Nealen et al. [2006] for an overview about the current state of the art.

Damping forces were already used by Terzopoulos [1988], who introduced a velocity-dependent damping force $\gamma\dot{\mathbf{x}}$ as in (2). [Platt and Barr 1988] proposed improved damping forces for Finite Elements which depend on the strain rate tensor. These were also used by [Carignan et al. 1992] and [O'Brien and Hodgins 1999], for example. Further, [Platt and Barr 1988] introduced damping forces for constrained motions that depend on the time-derivative of a point-to-nail constraint. [Baraff and Witkin 1998] generalized this kind of damping for arbitrary constraints $C(\mathbf{X})$ choosing damping forces that depend on $\dot{C}(\mathbf{X})$. A damping model which is independent of the deformation model, but relies on the connectivity of the object is described in [Nealen et al. 2006] and will be shown in Sec. 2.

In contrast to these approaches, our damping approach is based on an iterative damping scheme in order to further improve the stability. It is independent from the deformation model [Platt and Barr 1988; Baraff and Witkin 1998], the integration scheme and the structure of the simulated object [Nealen et al. 2006]. By choosing the damping parameter always smaller than 1, we can guarantee that our damping forces do not overshoot.

2 Spring damping approach

Before introducing our damping approach in Sec. 3, we briefly review the spring damping approach of, for example, [Nealen et al. 2006]. In this approach, a force is symmetrically applied to two adjacent points in order to damp their relative velocity.

As mentioned in Sec. 1, damping relative velocities improves the stability without changing the global movement. For the spring damping approach, some kind of neighborhood information for the points of an object is needed and damping forces are computed for each pair of points that are considered to be adjacent. Neighborhood can be defined, for example, by an edge in a tetrahedral mesh, or by an influence radius of a particle.

We denote the position of a mass point with $\mathbf{x}_i$ and its velocity with $\mathbf{v}_i$. A point is always referenced by its position. $\mathbf{f}_i$ denotes the force that acts on point $\mathbf{x}_i$ excluding damping, and $\mathbf{f}_i^d$ denotes the damping force at this point. Each pair of adjacent points in the neighborhood information is referenced by the distance or relative position $\mathbf{x}_e = \mathbf{x}_i - \mathbf{x}_j$ of the incident points. The relative velocity of two adjacent points is denoted as $\mathbf{v}_e = \mathbf{v}_i - \mathbf{v}_j$. Similar to the damping forces at the points, the damping forces which are related to relative velocities are given by $\mathbf{f}_e^d$.

The spring damping approach works as follows. For two points $\mathbf{x}_i$ and $\mathbf{x}_j$, the relative velocity $\mathbf{v}_e = \mathbf{v}_i - \mathbf{v}_j$ is computed. A damping force of $\mathbf{f}_e^d = -\gamma\mathbf{v}_e$, i. e. proportional to the relative velocity $\mathbf{v}_e$, is computed and symmetrically applied to $\mathbf{x}_i$ and $\mathbf{x}_j$. Since both damping forces add to zero, the linear momentum is preserved.

However, the angular momentum is generally not preserved. This issue can be addressed by projecting the forces onto the direction $\mathbf{x}_e$ using $\overline{\mathbf{f}_e^d} := (\mathbf{x}_e \cdot \mathbf{f}_e^d)\frac{\mathbf{x}_e}{\|\mathbf{x}_e\|_2^2}$.

The result of the computed damping forces can be further improved, if the future velocities $\mathbf{v}_i'$ and $\mathbf{v}_j'$ are predicted using an Euler integration step. In this case, the damping forces are computed with respect to the predicted future relative velocity $\mathbf{v}_e'$.

3 Direct computation of iterative spring damping forces

In this section, we explain our new damping approach. It is motivated by the fact that an iterative computation of spring damping forces within one simulation step can improve the reduction of oscillations. In the following, we show that the iterative damping procedure converges and that the limit can be computed directly. While the linear momentum is inherently preserved, we show and compare three different ways to guarantee that the angular momentum is preserved, too. Further, the parameter setting and the possibilities of application are shown.

Our approach employs the center of mass $\mathbf{x}_{cm}$ of an object and its velocity $\mathbf{v}_{cm}$. With $\mathbf{v}_i' = \mathbf{v}_i + \frac{h}{m_i}\mathbf{f}_i$, we denote the predicted future velocity of point $\mathbf{x}_i$, and similarly $\mathbf{v}_e'$ denotes the predicted future relative velocity of distance $\mathbf{x}_e$. With $\mathbf{v}_i^d = \mathbf{v}_i + \frac{h}{m_i}(\mathbf{f}_i + \mathbf{f}_i^d)$, we denote the damped velocity at point $\mathbf{x}_i$. The neighborhood information can be stored in the *connectivity matrix* $\mathbf{E} \in \mathbb{R}^{m\times n}$, where m is the number of adjacent point pairs in the neighborhood information and n is the number of points of the object. $\mathbf{E}$ is defined as follows: For a distance $\mathbf{x}_e$ with incident points $\mathbf{x}_i$ and $\mathbf{x}_j$, where $i < j$, we define $\mathbf{E}_{e,i} = 1$ and $\mathbf{E}_{e,j} = -1$. All other values of $\mathbf{E}$ are set to zero. For later use, $\mathbf{M}$ denotes a diagonal matrix containing the masses.

With $\mathbf{V} := (\mathbf{v}_1^T, \ldots, \mathbf{v}_n^T)^T$, we denote the set of current velocities, similarly with $\mathbf{V}'$ the predicted and with $\mathbf{V}^d$ the damped velocities. $\mathbf{F}$ denotes the set of forces, similarly $\mathbf{F}^d$ the damping forces, and $\mathbf{F}_e^d$ the set of the damping forces at the distances.

3.1 Iterative force computation

In this section, we illustrate how the spring damping forces can be computed iteratively. It is shown that this iterative force computation converges to a limit that can be easily computed. For this

procedure, we start with the standard spring damping step.

$$\begin{aligned}
\mathbf{v}_i' &= \mathbf{v}_i + \frac{h}{m_i}\mathbf{f}_i \\
\mathbf{v}_j' &= \mathbf{v}_j + \frac{h}{m_j}\mathbf{f}_j \\
\mathbf{v}_e' &= \mathbf{v}_i' - \mathbf{v}_j' \\
\mathbf{f}_e^d &= -\gamma \mathbf{v}_e' \\
\mathbf{f}_i^d &= \mathbf{f}_i^d + \mathbf{f}_e^d \\
\mathbf{f}_j^d &= \mathbf{f}_j^d - \mathbf{f}_e^d
\end{aligned}$$

For the damped velocities $\mathbf{v}_i^d$ and $\mathbf{v}_j^d$, we get

$$\begin{aligned}
\mathbf{v}_i^d &= \mathbf{v}_i + \frac{h}{m_i}(\mathbf{f}_i^d + \mathbf{f}_i) = \mathbf{v}_i' + \frac{h}{m_i}\mathbf{f}_i^d \\
\mathbf{v}_j^d &= \mathbf{v}_j + \frac{h}{m_j}(\mathbf{f}_j^d + \mathbf{f}_j) = \mathbf{v}_j' + \frac{h}{m_j}\mathbf{f}_j^d.
\end{aligned}$$

In the following part, we express the damping step in matrix-vector-notation using the connectivity matrix in order to show the convergence of this procedure.

$$\begin{aligned}
\mathbf{V}' &= \mathbf{V} + h\mathbf{M}^{-1}\mathbf{F} \\
\mathbf{F}_e^d &= -\gamma \widetilde{\mathbf{E}}\mathbf{V}' \\
\mathbf{F}^d &= \widetilde{\mathbf{E}}^T\mathbf{F}_e^d \\
&= -\gamma \widetilde{\mathbf{E}}^T\widetilde{\mathbf{E}}\mathbf{V}',
\end{aligned} \tag{4}$$

with $\widetilde{\mathbf{E}} \in \mathbb{R}^{3n\times 3n}$ arising from $\mathbf{E}$ by replacing each entry y of $\mathbf{E}$ by a 3×3-matrix $\begin{pmatrix} y & 0 & 0 \\ 0 & y & 0 \\ 0 & 0 & y \end{pmatrix}$. Further, we can compute the damped velocities as

$$\begin{aligned}
\mathbf{V}^d &= \mathbf{V}' + h\mathbf{M}^{-1}\mathbf{F}^d \\
&\overset{(4)}{=} \mathbf{V}' - \gamma h\mathbf{M}^{-1}\widetilde{\mathbf{E}}^T\widetilde{\mathbf{E}}\mathbf{V}' \\
\Rightarrow \mathbf{V}^d &= (\mathbf{id} - \gamma h\mathbf{M}^{-1}\widetilde{\mathbf{E}}^T\widetilde{\mathbf{E}})\mathbf{V}'
\end{aligned}$$

After k iterations we get

$$\begin{aligned}
\mathbf{V}^{d,k} &= (\mathbf{id} - \gamma h\mathbf{M}^{-1}\widetilde{\mathbf{E}}^T\widetilde{\mathbf{E}})^k\mathbf{V}' \\
&=: \mathbf{D}^k\mathbf{V}',
\end{aligned} \tag{5}$$

where $\mathbf{V}^{d,k}$ denotes the damped velocities after k iterations. For the convergence of (5), we observe that $\widetilde{\mathbf{E}}^T\widetilde{\mathbf{E}}$ is symmetric, hence diagonalizable, and positive semidefinite, as $\mathbf{x}^T\widetilde{\mathbf{E}}^T\widetilde{\mathbf{E}}\mathbf{x} = \|\widetilde{\mathbf{E}}\mathbf{x}\|_2^2 \geq 0$. Thus, its eigenvalues are greater than or equal to zero. The convergence of (5) now can be shown by discussing the eigenvalues of $\widetilde{\mathbf{E}}^T\widetilde{\mathbf{E}}$ and $\mathbf{D}$. If we can show that the absolute values of the eigenvalues are smaller than one, then (5) converges for $k \to \infty$.

If $\mathbf{v}_i' = \mathbf{v}_{cm}'$ for all points, we get $\widetilde{\mathbf{E}}\mathbf{V}' = 0$ and thus, $\mathbf{D}\mathbf{V}' = \mathbf{V}'$. Hence, 0 is always a triple eigenvalue of $\widetilde{\mathbf{E}}^T\widetilde{\mathbf{E}}$ as $\mathbf{v}_{cm}'$ has three possible directions, and 1 is always a triple eigenvalue of $\mathbf{D}$. Let $\mathbf{V}_{rel}'$ be the future velocity of the points relative to the center of mass. Then, we can partition $\mathbf{V}' = \mathbf{V}_{cm}' + \mathbf{V}_{rel}'$, which yields $\widetilde{\mathbf{E}}^T\widetilde{\mathbf{E}}\mathbf{V}' = \widetilde{\mathbf{E}}^T\widetilde{\mathbf{E}}\mathbf{V}_{rel}'$. That means, the damping forces (4) are not influenced by the velocity of the center of mass. This implies that we can damp only $\mathbf{V}_{rel}'$ and add $\mathbf{V}_{cm}'$ afterwards. Hence, we can replace $\mathbf{V}'$ in (5) by $\mathbf{V}_{rel}'$, which results in

$$\begin{aligned}
\mathbf{V}_{rel}^{d,k} &= (\mathbf{id} - \gamma h\mathbf{M}^{-1}\widetilde{\mathbf{E}}^T\widetilde{\mathbf{E}})^k\mathbf{V}_{rel}' \\
&= \mathbf{D}^k\mathbf{V}_{rel}'.
\end{aligned} \tag{6}$$

Thus, for the convergence of the damping forces it suffices to show that (6) converges. Hence, we can ignore the eigenvalue 1 of $\mathbf{D}$, which has no influence on (6) and have to care for the eigenvalue with the largest absolute value *besides* the eigenvalue 1, which we call the *key eigenvalue* in the following.

In the following Lemma, we show that the parameter of the spring damping approach can be chosen such that the absolute value of the key eigenvalue is strictly smaller than one.

Lemma 1. The damping parameter γ in (6) can always be chosen such that the absolute value of the key eigenvalue of $\mathbf{D}$ is strictly smaller than one.

Proof. The matrix $\widetilde{\mathbf{E}}^T\widetilde{\mathbf{E}}$ is positive semidefinite and symmetric, hence orthogonal diagonalizable. Further, its rank equals $3n-3$ as the rank of $\mathbf{E}$ is $n-1$ for a connected object. Thus, we know that exactly three eigenvalues are 0, which belong to the different directions of $\mathbf{V}_{cm}'$, and that all remaining eigenvalues are strictly greater than zero. Let $\mathbf{Q}$ be an orthogonal matrix with $\det(\mathbf{Q}) = 1$ such that $\Lambda := \mathbf{Q}^T\widetilde{\mathbf{E}}^T\widetilde{\mathbf{E}}\mathbf{Q}$ has diagonal form. The columns of $\mathbf{Q}$ can be identified as the eigenvectors of $\widetilde{\mathbf{E}}^T\widetilde{\mathbf{E}}$. Then $\mathbf{Q}^T\mathbf{D}\mathbf{Q} = \mathbf{Q}^T\mathbf{id}\mathbf{Q} - \gamma h\mathbf{M}^{-1}\mathbf{Q}^T\widetilde{\mathbf{E}}^T\widetilde{\mathbf{E}}\mathbf{Q} = \mathbf{id} - \gamma h\mathbf{M}^{-1}\Lambda$ has diagonal form, too. Thus, the eigenvalues $\mu_i,\ i = 1,\ldots,n$ of $\mathbf{D}$ can be written as $1-\lambda_i,\ i = 1,\ldots,n$, with λ_i being the eigenvalues of $\gamma h\mathbf{M}^{-1}\widetilde{\mathbf{E}}^T\widetilde{\mathbf{E}}$.

Obviously, we can choose γ dependent on $\mathbf{M}$ and h small enough such that all eigenvalues of $\gamma h\mathbf{M}^{-1}\mathbf{E}^T\mathbf{E}$ besides 0 lie in the interval $(0,2)$. This immediately yields $\mu_i \in (-1,1)$ for all eigenvalues of $\mathbf{D}$ besides 1, and we see that the absolute value of the key eigenvalue is strictly smaller than one. □

Based on Lemma 1, we can prove the convergence of (6).

Theorem 1. If γ is chosen small enough in (6), then $\mathbf{V}_{rel}^{d,k} \to 0$ for $k \to \infty$.

Proof. In Lemma 1, we showed that γ can be chosen such that the absolute value of the key eigenvalue μ is strictly smaller than one. Thus, it follows that $\|\mathbf{V}_{rel}^{d,k}\| \leq \mu\|\mathbf{V}_{rel}^{d,k-1}\| \leq \ldots \leq \mu^k\|\mathbf{V}_{rel}'\|$. This yields $\mathbf{V}_{rel}^{d,k} \to 0$ for $k \to \infty$. □

3.2 Direct force computation

In Theorem 1, we showed that the iterative spring damping approach leads to $\mathbf{V}_{rel}^d = 0$ for an appropriately chosen γ. Of course, the convergence needs lots of iterations and therefore, it is much less efficient than the simple spring damping approach. However, the limit of the damping forces can be computed directly using Theorem 1. We simply have to ensure that the computed damping forces yield $\mathbf{V}_{rel}^d = 0$. For a single point, this reads $\mathbf{v}_{i,rel}^d = \mathbf{v}_{i,rel}' + \frac{h}{m_i}\mathbf{f}_i^d = 0$, and the damping forces can be computed as

$$\mathbf{f}_i^d = -\frac{m_i}{h}\mathbf{v}_{i,rel}'. \tag{7}$$

This computation is very simple and efficient, and it is justified by the fact that it is the limit of an infinite number of iterations of the standard spring damping approach.

It is easy to introduce a damping parameter γ in (7) to compute the damping forces as

$$\mathbf{f}_i^d = -\gamma\mathbf{v}_{i,rel}', \tag{8}$$

which obviously should not be greater than one. Hence, γ is always within the range between 0 and 1.

Note that we did not use any specific information about the structure of $\widetilde{\mathbf{E}}$ in the proof of Theorem 1 except the fact that its rank is $3n-3$. But this is correct if the rank of $\mathbf{E}$ is $n-1$, which is fulfilled if the object is connected. Thus, the proof holds for any type of connectivity structure that defines a connected object. Further, the connectivity information canceled out in the limit for $k \to \infty$ in (6), and as seen in (8), it is not needed any more.

3.3 Momentum conservation

As damping forces should not influence the global movement of an object, they must guarantee the conservation of linear and angular momentum. While it is easy to show that the sum of the damping forces in (8) is zero and the linear momentum is conserved, the computation of damping forces commonly results in a nonzero torque, i. e. the condition

$$\sum_{i=1}^{n} (\mathbf{r}_i \times \mathbf{f}_i^d) = 0, \tag{9}$$

where $\mathbf{r}_i = \mathbf{x}_i - \mathbf{x}_{cm}$, is not fulfilled, which has to be handled in a post-processing step.

First, we show the conservation of the linear momentum:

$$\begin{aligned}
\sum_{i=1}^{n} \mathbf{f}_i^d &= -\sum_{i=1}^{n} \frac{m_i}{h} \mathbf{v}'_{i,rel} \\
&= -\frac{1}{h} \sum_{i=1}^{n} m_i (\mathbf{v}'_i - \mathbf{v}'_{\mathbf{cm}}) \\
&= -\frac{1}{h} \left(\sum_{i=1}^{n} m_i \mathbf{v}'_i - \mathbf{v}'_{cm} \sum_{i=1}^{n} m_i \right) \\
&= 0.
\end{aligned}$$

Now, we present our ideas how the torque can be eliminated. In the simple spring damping approach, damping forces are computed per distance. To cancel out the torque, they can simply be projected onto the distance, which results in zero torque.

The new damping approach in (8) computes forces per point instead of forces per distance. Therefore, we tried out three new ideas to reduce the torque which are presented in the following.

3.3.1 Force projection onto the distances

The first method is similar to the simple spring damping approach, as we transform the forces per point into forces per distance and project them. This is done the following way.

For a distance $\mathbf{x}_e$ with incident points $\mathbf{x}_i$ and $\mathbf{x}_j$, we set $\mathbf{f}_e^d = \mathbf{f}_i^d - \mathbf{f}_j^d$ and afterwards, we project this force onto $\mathbf{x}_e$ using $\overline{\mathbf{f}_e^d} = (\mathbf{f}_e^d \cdot \mathbf{x}_e) \frac{\mathbf{x}_e}{\|\mathbf{x}_e\|_2^2}$ like in Sec. 2.

After projecting the force, we distribute $\overline{\mathbf{f}_e^d}$ to the incident points using a temporary variable $\overline{\mathbf{f}_i^d}$ and $\overline{\mathbf{f}_j^d}$:

$$\begin{aligned}
\overline{\mathbf{f}_i^d} &= \overline{\mathbf{f}_i^d} + \overline{\mathbf{f}_e^d} \\
\overline{\mathbf{f}_j^d} &= \overline{\mathbf{f}_j^d} - \overline{\mathbf{f}_e^d}.
\end{aligned}$$

This obviously results in zero torque. Having processed all distances $\mathbf{x}_e$, the magnitude of the force $\overline{\mathbf{f}_i^d}$ can be much higher than the magnitude of $\mathbf{f}_i^d$. To overcome this problem, we have to renormalize the damping forces with a constant factor c which is given by

$$c = \min_{i=1,\dots,n} \frac{\|\mathbf{f}_i^d\|}{\|\overline{\mathbf{f}_i^d}\|} \tag{10}$$

and set $\mathbf{f}_i^d = c\overline{\mathbf{f}_i^d}$.

3.3.2 Force projection onto other relative positions

If we have a particular point $\mathbf{x}$ in our connectivity structure, for example the center of mass, it can be more suitable to project the forces onto the relative position $\mathbf{r}_i^x := \mathbf{x}_i - \mathbf{x}$. Obviously, the torque with respect to $\mathbf{x}$, i. e. $\sum_{i=1}^n \mathbf{r}_i^x \times \mathbf{f}_i^d$, is zero, as $\mathbf{r}_i^x$ and $\mathbf{f}_i^d$ are collinear and thus, their cross product is zero. However, after this projection the sum of the forces is not necessarily zero. This can be handled the following way: Let $\mathbf{x} = \sum \alpha_i \mathbf{x}_i$ with $\sum \alpha_i = 1$, and $\sum \mathbf{f}_i^d = \mathbf{f}$. If we subtract $\alpha_i \mathbf{f}$ from each $\mathbf{f}_i^d$, the sum of the forces will be zero again. We claim that the torque remains zero, too. This is indeed true:

$$\begin{aligned}
\sum_{i=1}^{n} \mathbf{r}_i^x \times (\alpha_i \mathbf{f}) &= \sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{x}) \times (\alpha_i \mathbf{f}) \\
&= \sum_{i=1}^{n} (\alpha_i \mathbf{x}_i) \times \mathbf{f} - \mathbf{x} \times \mathbf{f} \sum_{i=1}^{n} \alpha_i \\
&= \mathbf{x} \times \mathbf{f} - \mathbf{x} \times \mathbf{f} = 0,
\end{aligned}$$

and we end up with zero torque and zero force sum. Note that the last force correction does not impose additional relative movement onto the affected nodes and therefore, the damping is only affected by the projection which justifies this procedure. The linear momentum is not affected at all. We will discuss the application of this projection approach in Sec. 4.

3.3.3 Torque elimination using Linear Programming

We also implemented a method of torque elimination using Linear Programming. Before we establish the Linear Program, we remind that a cross product can be written as a matrix-vector-multiplication: For a vector $\mathbf{a} = (a_x, a_y, a_z)^T$, we define a skew-symmetric matrix

$$\widetilde{\mathbf{A}} := \begin{pmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{pmatrix}.$$

Then, the cross product $\mathbf{a} \times \mathbf{b}$ can be written as $\widetilde{\mathbf{A}}\mathbf{b}$. The zero torque condition (9) then reads

$$\sum_{i=1}^{n} \widetilde{\mathbf{R}}_i \mathbf{f}_i^d = 0.$$

As this sum is not zero in general, we have to compute correction forces $\mathbf{f}_i^c$ to cancel out the torque. Then the condition becomes

$$\sum_{i=1}^{n} \widetilde{\mathbf{R}}_i (\mathbf{f}_i^d + \mathbf{f}_i^c) = 0, \tag{11}$$

which is one of the conditions for the Linear Program.

The sum of the correction forces $\sum_{i=1}^{n} \mathbf{f}_i^c$ has to be zero, because $\sum_{i=1}^{n} \mathbf{f}_i^d$ is zero. Further, we demand that the correction forces should be as small as possible. Thus, the objective function has to be:

$$\min \sum_{i=1}^{n} \|\mathbf{f}_i^c\|_1. \tag{12}$$

For a Linear Programming Problem, we have to eliminate the absolute value from the objective function. This can be done by partitioning $\mathbf{f}_i^c = \mathbf{f}_i^{c,+} - \mathbf{f}_i^{c,-}$ into its positive and negative components together with the constraint that $\mathbf{f}_i^{c,+}$ and $\mathbf{f}_i^{c,-}$ are greater than or equal to zero:

$$\begin{aligned} &\min \sum_{i=1}^{n} \|\mathbf{f}_i^{c,+} - \mathbf{f}_i^{c,-}\|_1 \\ =&\min \sum_{i=1}^{n} (\|\mathbf{f}_i^{c,+}\|_1 + \|\mathbf{f}_i^{c,-}\|_1). \end{aligned} \tag{13}$$

Eq. (13) holds, as it is a minimization problem: If, for example, the first component of $\mathbf{f}_i^{c,+}$ and the first component of $\mathbf{f}_i^{c,-}$ were both nonzero, this would not be the optimal solution of the minimization problem. Thus, for the optimal solution, always at least one of the corresponding components of $\mathbf{f}_i^{c,+}$ and $\mathbf{f}_i^{c,-}$ must be zero, implying the correctness of (13). Note that (13) does not contain absolute values for the components of $\mathbf{f}_i^{c,+}$ and $\mathbf{f}_i^{c,-}$, as we demanded that $\mathbf{f}_i^{c,+}$ and $\mathbf{f}_i^{c,-}$ are greater than or equal zero.

In total, the Linear Program for torque reduction can be established as follows:

$$\begin{aligned} &\min \sum_{i=1}^{n} (\mathbf{f}_i^{c,+} + \mathbf{f}_i^{c,-}) \\ s.t. \quad &\mathbf{f}_i^{c,+} \geq 0 \\ &\mathbf{f}_i^{c,-} \geq 0 \\ &\sum_{i=1}^{n} \widetilde{\mathbf{R}}_i (\mathbf{f}_i^d + \mathbf{f}_i^{c,+} - \mathbf{f}_i^{c,-}) = 0 \\ &\sum_{i=1}^{n} (\mathbf{f}_i^{c,+} - \mathbf{f}_i^{c,-}) = 0 \end{aligned} \tag{14}$$

This Linear Program is feasible, as $\mathbf{f}_i^{c,+} - \mathbf{f}_i^{c,-} = -\mathbf{f}_i^d$ is a possible solution, and the objective function is always greater than or equal zero. Thus, it has an optimal solution which eliminates the torque.

3.4 Application

In this part, we describe two example settings that show how the new damping approach can be applied to compute damping forces. We refer to these methods as global and local damping, as we damp the whole object in the first one and smaller structures locally in the second one. The influence of the torque elimination method on these settings will be discussed in Sec. 4.

3.4.1 Global damping

The global damping approach is the most evident idea following (8). In this version, we damp the object as a whole, so we compute the center of mass of the object, calculate the damping forces as given in (8) and eliminate the torque with one of the above mentioned ideas.

3.4.2 Local damping

In this approach, we divide the object into clusters and damp the movement of each cluster relative to its center of mass, but not relative to the center of mass of the whole object. For example, in a tetrahedral mesh we could simply think of the tetrahedrons as clusters. To get reasonable damping forces that do not overshoot, we have to care for the number of clusters a point lies in. Therefore, we divide the mass m_i of point $\mathbf{x}_i$ by the number of clusters N_i the point belongs to and use m_i/N_i as the mass of the $\mathbf{x}_i$ within each cluster (cf. [Rivers and James 2007]).

Of course, it is possible to first perform a global damping step which stabilizes the object in the case of external forces and to perform a local damping step afterwards to reduce oscillations.

4 Discussion

In this section, we analyze the properties of the damping approach and the different schemes to eliminate the torque.

Obviously, the damping forces (7) are optimal in the sense that they damp the whole relative movement. Therefore, they lead to an unconditionally stable simulation. But as they lead to a nonzero torque, it is necessary to handle this drawback. Eliminating the torque obviously introduces some uncontrollable effects, which should be minimized by the elimination technique.

The three different approaches that we proposed to eliminate the torque differ in an essential fact. While the first projection technique, which uses the distances, cares for a local torque elimination, the second projection approach and the LP formulation eliminate the torque only globally. Also, they guarantee the conservation of the global linear momentum, while projecting onto the distances preserves global and local linear momentum. Our experiments showed that the global conservation of linear and angular momentum is not strong enough, as forces and torques at one part of an object can be balanced by forces and torques at another part. This results in undesired artifacts.

Another difference of the approaches is that the projecting methods do not look for a minimum solution, but instead they choose a specific direction. As both methods show better results than the LP formulation, it seems that it is much more important to have any control about the directions of the projected forces than to minimize the magnitude of the correction forces. This is further affirmed by the results we got when we applied the LP formulation for each tetrahedron locally.

Following this, for the global damping approach, projecting the forces onto the distances currently is the best technique to eliminate the torque, as it guarantees the local conservation of linear and angular momentum. Concerning local damping, we propose to use the second projection method, as the directions from the center of mass to the points are more likely to match the directions of the damping forces which are also relative to the center of mass. The locality of momentum conservation is fulfilled self-evidently in the local damping.

The projection methods can be applied to any set of forces that should guarantee the preservation of linear and angular momentum. For example, they could be applied to constraint forces in order to preserve the global movement of an object. Even the point damping forces would yield reasonable damping forces after applying the torque elimination procedure.

For the local damping approach, it turned out that omitting the projection results in great improvement of stability for the drawback that the global orientation is affected. However, there are scenarios where this fact is accepted because of the great improvement of stability.

5 Results

We implemented various scenes to show the benefits of damping in dynamic simulations and to illustrate the abilities of our damping approach. First, we show the capabilities of the spring damping approach (Fig. 1), and afterwards, we show the improvements achieved by our method (Fig. 2 and 3). All objects consist of tetrahedral meshes generated by [Spillmann et al. 2006].

In Fig. 1, we illustrate the differences between an undamped, a point-damped and a spring-damped simulation. We applied external forces to the object using a spring dragger. While the undamped object suffers from local oscillations and the simulation finally fails (Fig. 1 (a)), the point-damped one comes to a resting state quickly, but it does not fall down as the global movement is disturbed (Fig. 1 (b)). In contrast, the spring-damped sneaker (Fig. 1 (c,d)) behaves perfectly and keeps stable even with a significantly enlarged time step.

In Fig. 2, we illustrate the ability of our damping approach to handle scenes where the spring damping approach cannot keep the simulation stable. Here, a global damping step is performed which is followed by a local damping step. This combination yielded the best results. Note that before we perform the local damping, the global damping forces are already projected using the first projection method. The setting of the experiment is quit simple: We let a sphere fall down onto a membrane. While the simulation fails using the spring damping approach, it remains perfectly stable with our approach. Collisions are detected using [Teschner et al. 2003], and contacts are handled using [Heidelberger et al. 2004].

Fig. 3 illustrates another scenario where our approach allows for a larger time step compared to existing solutions. Like in Fig. 2, we applied both a global and a local damping step. The rope bridge consists of many tetrahedral meshes that are connected with local constraints [Gissler et al. 2006]. Due to the locality and magnitude of the constraint forces, a fast force propagation is very important to keep the object stable, which is implicitly done by our damping approach. This also holds for any type of external forces.

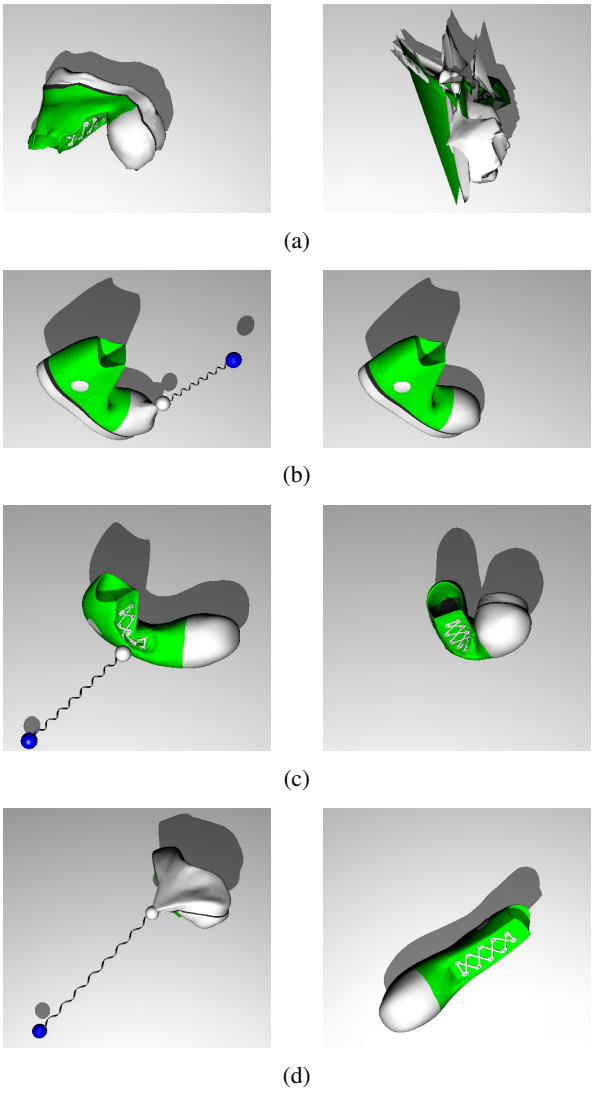

Figure 1: (a) The undamped sneaker does not recover to a stable resting state. (b) The point-damped sneaker remains stable and recovers to a resting state quickly. However, the global movement is heavily influenced. (c,d) The spring-damped sneaker recovers to a stable resting state. The global movement is not affected.

6 Conclusion

In this paper, we introduced a new damping approach that is inspired by the idea that an iterative computation of damping forces further improves the stability. We showed that the iterative spring damping approach converges, if the damping parameter is chosen appropriately, and that the limit can be computed directly without actually performing iterations. The approach is independent of the deformation model and the integration scheme and does not need connectivity information. Thus, it can be applied for arbitrary object representations, e. g. meshless approaches. While the linear momentum is automatically conserved, we proposed different techniques to eliminate the torque. These approaches can also be applied to any other set of forces that have to guarantee the conservation of angular momentum. We illustrated the possibilities of our approach to be used as global damping to the object as a whole or as local damping to separate clusters of the object. Further, we showed that our approach simplifies the parameter setting, as the damping constant is always within the range between 0 and 1. In the result section, we showed that our approach allows for larger time steps in dynamic simulations. Ongoing work is concerned with improved torque elimination techniques to further reduce the influence on the damping and with the possible application of the force projection techniques in different contexts.

Acknowledgments

This project is supported by the German Research Foundation (DFG) under contract number SFB TR8 and by the AO Foundation. We thank the reviewers for their helpful comments.

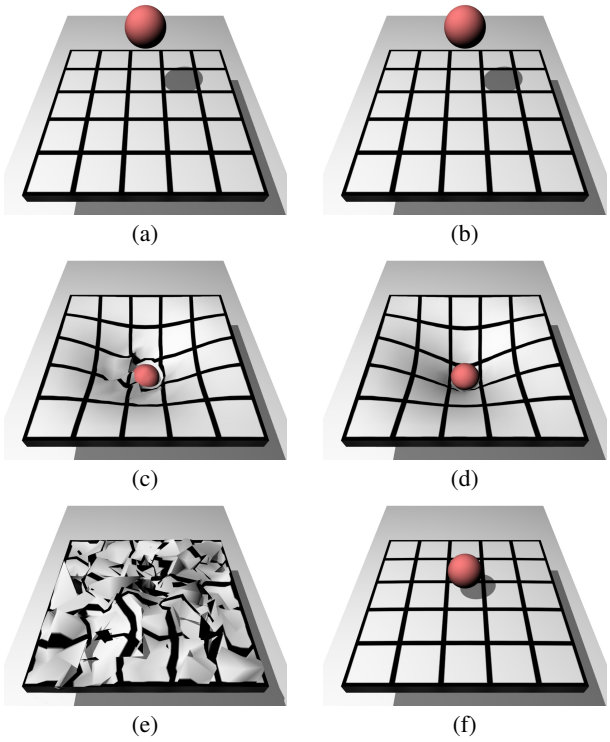

Figure 2: This experiment illustrates the differences between the spring damping approach (left column) and our approach (right column). The compared pictures are always chosen at the same simulation step. (a,b) Configuration of the experiment: A sphere falls down onto the membrane. (c) Using the spring damping approach, the membrane does not remain stable due to the response forces. (d) The membrane shows no artifacts using our approach. (e) After the collision contact, the membrane does not return to its resting state using the spring damping. (f) The sphere bounces back using our approach.

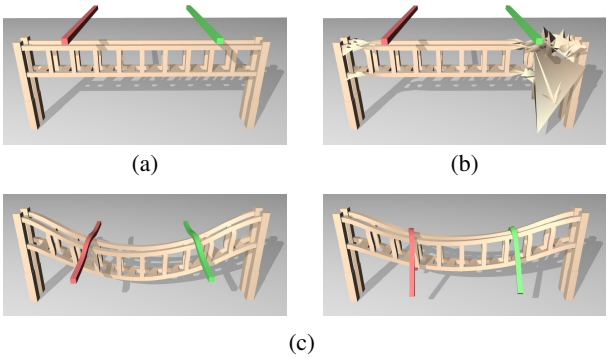

Figure 3: This experiment illustrates a scenario where a larger time step can be chosen using our approach. (a) Setting of the experiment: Two elastic bars fall onto a rope bridge. (b) Spring damping approach: Even before collision contact, the bridge cannot be simulated stably. The simulation fails within few simulation steps. (c) Using our approach, the bridge remains stable and the collisions are handled robustly.

References

BARAFF, D., AND WITKIN, A. 1998. Large steps in cloth simulation. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 43–54.

BOURGUIGNON, D., AND CANI, M.-P. 2000. Controlling anistotropy in mass-spring systems. In *Proc. Computer Animation and Simulation*, 113–123.

CARIGNAN, M., YANG, Y., THALMANN, N. M., AND THALMANN, D. 1992. Dressing animated synthetic actors with complex deformable clothes. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 99–104.

CHEN, Y., ZHU, Q., KAUFMANN, A., AND MURAKI, S. 1998. Physically-based animation of volumetric objects. In *Proc. of IEEE Computer Animation*, 154–160.

GISSLER, M., BECKER, M., AND TESCHNER, M. 2006. Local constraint sets for deformable objects. In *Proc. Virtual Reality Interactions and Physical Simulations VRIPHYS*, 25–32.

HAUTH, M., AND STRASSER, W. 2004. Corotational simulation of deformable solids. *Journal of WSCG 12*, 137–145.

HEIDELBERGER, B., TESCHNER, M., KEISER, R., MÜLLER, M., AND GROSS, M. 2004. Consistent penetration depth estimation for deformable collision response. In *Proc. Vision, Modeling, Visualization VMV*, 339–346.

MÜLLER, M., AND GROSS, M. 2004. Interactive virtual materials. In *Proceedings of Graphics Interface*, 239–246.

MÜLLER, M., DORSEY, J., MCMILLAN, L., JAGNOW, R., AND CUTLER, B. 2002. Stable real-time deformations. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM, New York, NY, USA, 49–54.

MÜLLER, M., HEIDELBERGER, B., TESCHNER, M., AND GROSS, M. 2005. Meshless deformations based on shape matching. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, ACM, New York, NY, USA, 471–478.

NEALEN, A., MÜLLER, M., KEISER, R., BOXERMAN, E., AND CARLSON, M. 2006. Physically based deformable models in computer graphics. *Computer Graphics Forum 25*, 4, 809–836.

O'BRIEN, J. F., AND HODGINS, J. K. 1999. Graphical modeling and animation of brittle fracture. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 137–146.

PLATT, J. C., AND BARR, A. H. 1988. Constraints methods for flexible models. In *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 279–288.

RIVERS, A. R., AND JAMES, D. L. 2007. FastLSM: Fast lattice shape matching for robust real-time deformation. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, ACM, New York, NY, USA, 82.

SPILLMANN, J., WAGNER, M., AND TESCHNER, M. 2006. Robust tetrahedral meshing of triangle soups. In *Proc. Vision, Modeling, Visualization VMV*, 9–16.

TERZOPOULOS, D., AND FLEISCHER, K. 1988. Deformable models. *The Visual Computer 4*, 306–331.

TERZOPOULOS, D., PLATT, J., BARR, A., AND FLEISCHER, K. 1987. Elastically deformable models. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 205–214.

TESCHNER, M., HEIDELBERGER, B., MÜLLER, M., POMERANETS, D., AND GROSS, M. 2003. Optimized spatial hashing for collision detection of deformable objects. In *Proc. Vision, Modeling, Visualization VMV*, 47–54.

TESCHNER, M., HEIDELBERGER, B., MÜLLER, M., AND GROSS, M. 2004. A versatile and robust model for geometrically complex deformable solids. In *Proc. Computer Graphics International*, 312–319.