# Integrity Improvements in Classically Deformable Solids

Micky K. Christensen* Anders Fleron†

Department of Computer Science
University of Copenhagen
DIKU

## Abstract

The physically-based model for simulating elastically deformable objects, presented by Terzopoulos et al. in 1987, has significant problems with handling solids. We give a plausible explanation for the reasons of instability that lead to implosion of deformable solids. We propose an extension to the original model, with improvements that result in a model of increased stability. Comparisons are made with the original method to illustrate this point. The improved model is suitable for interactive simulations of deformable solids, with only a small constant decrease of performance.

**Keywords:** Physics-based animation, simulation, dynamics, deformation, real time, deformable solids, continuum elastic model, constraints, variational calculus, finite differences

## 1 Introduction

Deformable models seem to have gained increasing interest during the last years. Part of this success comes from a desire to interact with objects that resembles those in real life, which all seem to be deformable at some level. The fact that CPUs and GPUs today are both advanced and powerful makes it possible to simulate and animate deformable bodies interactively. This paper builds on work previously done by Christensen and Fleron [1], which focused on an implementation of a method for simulating elastically deformable objects, first put forward by Terzopoulos et al. in 1987 [9]. The implementation could simulate various types of deformable surfaces convincingly, but severable instability issues were noticed, with regard to simulating deformable solids. As the authors of the original method claim the model can simulate deformable curves, surfaces, and solids, we will look into why the method was unable to simulate deformable solids satisfactorily. At first sight it seemed that the solids did have a hard time keeping their integrity. Integrity preservation is important to give a realistic impression of a deformable solid. We will try to solve the instabilities using concepts from the original frame work, and propose an improved model that is able to simulate solids, as well as surfaces, without a significant decrease of the overall performance.

In section 2 we revisit the theory of elastically deformable models, with focus on solids, to give an overview of the method. The theory serves as a foundation for understanding the following sections. Section 3 reveals and explains the instabilities of the original model. Sections 4 and 5 focus on solutions and improvements of the local and global problems, respectively. In section 6 we present the results of adding the developed improvements to the model. Comparison between the original and improved model will be performed visually.

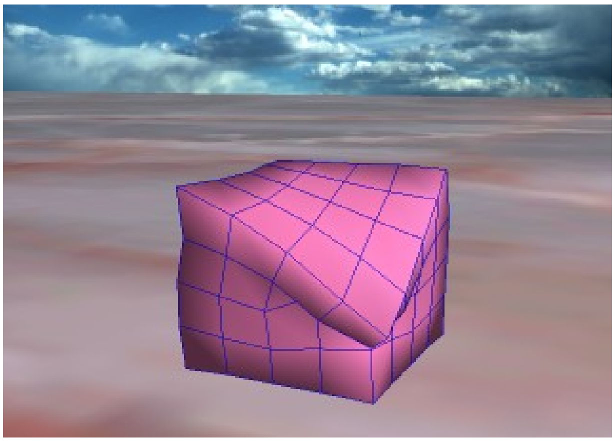

**Figure 1** Large deformation results in convincing material buckling

## 2 Elastically Deformable Solids

At the basis of the theory of deformable models lies elasticity theory. From this physical theory Terzopoulos et al. have extrapolated a model that governs the movements of elastically deformable bodies [9]. In this paper we will concentrate on the theory of deformable solids.

A point in a solid is described by the intrinsic coordinates $\mathbf{a} = [a_1, a_2, a_3]$. A deformable solid is thought of as having a natural rest state, where no elastic energy is inherent. The rest state is described by $\mathbf{r}^0(\mathbf{a}) = [r_1^0(\mathbf{a}), r_2^0(\mathbf{a}), r_3^0(\mathbf{a})]$, where the positional vector function $\mathbf{r}$ of the object is defined in Euclidian 3-space. When the solid is deformed, it takes on a different shape than its rest shape, and distances between nearby points are either stretched or compressed with the deformation. This ultimately creates elastic energy that results in internal forces that will seek to minimize the elasticity.

The deformation will evolve over time and as such, it can be described by the time-varying vector function $\mathbf{r}(\mathbf{a},t) = [r_1(\mathbf{a},t), r_2(\mathbf{a},t), r_3(\mathbf{a},t)]$. The evolving deformation is independent of the rigid body motion of the solid. The equations governing the motion of particles in a deformable solid are obtained from Newtonian mechanics, and given by

$$\frac{\partial}{\partial t}\left(\mu \frac{\partial \mathbf{r}}{\partial t}\right) + \gamma \frac{\partial \mathbf{r}}{\partial t} + \frac{\delta \varepsilon(\mathbf{r})}{\delta \mathbf{r}} = \mathbf{f}(\mathbf{r},t), \qquad (1)$$

[*] micky@mkc.dk
[†] afleron@knights.dk

where $\mathbf{r}(\mathbf{a},t)$ is the position of the particle $\mathbf{a}$ at time $t$, $\mu(\mathbf{a})$ is the mass density at particle $\mathbf{a}$, and $\gamma(\mathbf{a})$ is the damping density. The right hand side represents the sum of externally applied forces at time $t$. The third term on the left hand side of (1) is called a variational derivative, and this is where the internal elastic energy is dealt with. $\varepsilon(\mathbf{r})$ is called a functional and it measures the potential energy that builds up when the body is deformed.

### 2.1 Deformation Energy

A method is needed to measure the deformation energies that arise when a solid deforms. For this task, we use the area of differential geometry. It is convenient to look at arc-lengths on curves, running along the intrinsic directions of the solid. A way of measuring the directions is specified by what is known as the first fundamental form. This form takes on a local view, and looks at changes in length between particles in a small neighborhood of a particle. When looking at the first fundamental form, or the metric tensor, we get an idea of the distances between particles. For a deformable solid the metric tensor is

$$G_{ij}(\mathbf{r}(\mathbf{a})) = \frac{\partial \mathbf{r}}{\partial a_i} \cdot \frac{\partial \mathbf{r}}{\partial a_j}, \quad 1 \le i,j \le 3\,, \tag{2}$$

which is a symmetric $3 \times 3$ tensor. The diagonal of the tensor represents length measurements along the coordinate directions from the particle in question. The off-diagonal elements represent angle measurements between the coordinate directions, which resist shearing within the local particle structure.

When measuring deformation energy in a solid, we are interested in looking at the change of the shape, with respect to the natural rest shape. This rest shape is described by the superscript $0$ such that

$$G_{ij}^0(\mathbf{r}(\mathbf{a})) = \frac{\partial \mathbf{r}^0}{\partial a_i} \cdot \frac{\partial \mathbf{r}^0}{\partial a_j}\,. \tag{3}$$

By using the weighted Hilbert-Schmidt matrix norm of the difference between the metric tensors, a simplified way of describing the energy of deformation becomes

$$\varepsilon(\mathbf{r}) = \int_\Omega \sum_{i,j=1}^{3} \eta_{ij}\left(G_{ij} - G_{ij}^0\right)^2 da_1 da_2 da_3\,, \tag{4}$$

where $\Omega$ is the domain of the deformable solid and $\boldsymbol{\eta}$ is a user defined tensor that weights each of the coefficients of the metric. For a single point the energy function is simply

$$S = \sum_{i,j=1}^{3} \eta_{ij}\left(G_{ij} - G_{ij}^0\right)^2\,. \tag{5}$$

The elastic energies that occur when a solid is deformed come from the stretching and compressing of the particles in the body. To discover the natural behavior of a deformed solid seeking towards its rest state, a term that minimizes the deformation energy is desirable. Variational calculus can be applied to find such a minimizing term, which is described by the Euler-Lagrange differential equations [5]. For a problem defined by a function of three independent variables and their first derivatives, the equations become

$$\frac{\partial F}{\partial \overline{y}} - \frac{d}{dx_1}\frac{\partial F}{\partial \overline{y}_{x_1}} - \frac{d}{dx_2}\frac{\partial F}{\partial \overline{y}_{x_2}} - \frac{d}{dx_3}\frac{\partial F}{\partial \overline{y}_{x_3}} = 0\,. \tag{6}$$

The symbol $F$ is the function describing the problem, and $F$ will be minimized with respect to the symbol $\overline{y}$, which in our case is the positional vector field $\mathbf{r}$. Substituting (5) into $F$ in (6) and executing the differentiations yields a neatly compact expression that minimizes the given problem

$$\frac{\delta S}{\delta \mathbf{r}} = -\sum_{i,j=1}^{3} \partial_{a_i}\left(\alpha_{ij}\mathbf{r}_{a_j}\right), \tag{7}$$

with

$$\alpha_{ij} = \eta_{ij}\left(\mathbf{r}_{a_i} \cdot \mathbf{r}_{a_j} - G_{ij}^0\right). \tag{8}$$

The alpha tensors (8) represent the comparison between the deformed state and the rest state of the solid. When an element in an alpha tensor becomes positive, it means that the corresponding constraint has been stretched and that it wants to shrink. Likewise, when an element becomes negative, the constraint has been compressed and it wants to grow.

### 2.2 Discretization

The deformable model is continuous in the intrinsic coordinates. To allow an implementation of deformable solids, the model is discretized into a unit 3D grid structure, representing the particles which will make up a solid. The grid will have three principal directions called $l$, $m$, and $n$. Particles in the grid are uniformly distributed with spacings in each of the three directions, given by the symbols $h_1$, $h_2$, and $h_3$. The number of particles in each of the directions are designated $L$, $M$, and $N$. Each particle property will be discretized using an index $[l,m,n]$, which will return the property value of that particular grid entry, e.g. the particle positions will be described by the discrete vector function $\mathbf{r}[l,m,n]$.

The model requires that derivatives are calculated in the intrinsic directions of the object. For this purpose we use finite difference operations across the grid, to achieve the desired derivative approximations [2]. An approximation to the first derivative in the $m$-direction, can for example be obtained by use of either a forward or backward difference operator. Given an arbitrary grid function $\mathbf{u}[l,m,n]$ these approximations are

$$D_2^+(\mathbf{u}) = h_2^{-1}\left(\mathbf{u}[l,m+1,n] - \mathbf{u}[l,m,n]\right) \tag{9}$$

and

$$D_2^-(\mathbf{u}) = h_2^{-1}\left(\mathbf{u}[l,m,n] - \mathbf{u}[l,m-1,n]\right), \tag{10}$$

where the superscript symbols, $+$ and $-$, represent forward and backward operators, respectively.

Using the difference operators, it is possible to discretize (7), by replacing the derivatives with the corresponding difference operators. The discrete equation for the elastic force $\mathbf{e}$ becomes

$$\mathbf{e}[l,m,n] = \sum_{i,j=1}^{3} -D_i^-(\mathbf{p})[l,m,n]\,, \tag{11}$$

$$\mathbf{p}[l,m,n] = \alpha_{ij}[l,m,n] D_j^+(\mathbf{r})[l,m,n]\,, \tag{12}$$

and the $\boldsymbol{\alpha}$ tensor field is also discretized using finite differencing,

$$\begin{aligned} &\alpha_{ij}[l,m,n] = \\ &\eta_{ij}[l,m,n](D_i^+(\mathbf{r})[l,m,n] \cdot D_j^+(\mathbf{r})[l,m,n] \\ &\qquad - G_{ij}^0[l,m,n]). \end{aligned} \tag{13}$$

Problems at the boundaries of the grid will occur because no information is readily available about the derivatives in these places. A natural boundary condition can be created by setting to zero all forward difference operators that reach outside the grid [9].

To solve equations for all particles at the same time, the values in the positional grid $\mathbf{r}$ and in the energy grid $\mathbf{e}$ can be unwrapped into $LMN$-dimensional vectors $\underline{\mathbf{r}}$ and $\underline{\mathbf{e}}$. With these vectors, the entire system of equations can be written as

$$\underline{\mathbf{e}} = \mathbf{K}(\underline{\mathbf{r}})\underline{\mathbf{r}}\,, \tag{14}$$

where $\mathbf{K}(\underline{\mathbf{r}})$ is an $LMN \times LMN$ sized matrix, called the stiffness matrix, which has desirable computational properties such as sparseness and bandedness. A discussion on how to assemble the stiffness matrix $\mathbf{K}$ can be found in [1].

We introduce the diagonal $LMN \times LMN$ mass matrix $\mathbf{M}$ and damping matrix $\mathbf{C}$, assembled from the corresponding discrete values of $\mu[l,m,n]$ and $\gamma[l,m,n]$, respectively. The equations of the elastically deformable model (1) can now be expressed in grid vector form, by the coupled system of second order ordinary differential equations

$$\mathbf{M}\frac{\partial^2 \underline{\mathbf{r}}}{\partial t^2} + \mathbf{C}\frac{\partial \underline{\mathbf{r}}}{\partial t} + \mathbf{K}(\underline{\mathbf{r}})\underline{\mathbf{r}} = \underline{\mathbf{f}}\,. \tag{15}$$

### 2.3 Numerical Integration

To evolve the deformable model over time, a semi-implicit time integration scheme will be employed. The time interval that the model will evolve in, is subdivided into time steps of equal size $\Delta t$. Using central differences, the time derivatives are approximated by

$$\begin{aligned} \frac{\partial^2 \underline{\mathbf{r}}}{\partial t^2} &\approx \frac{\underline{\mathbf{r}}_{t+\Delta t} - 2\underline{\mathbf{r}}_t + \underline{\mathbf{r}}_{t-\Delta t}}{\Delta t^2} \\ \frac{\partial \underline{\mathbf{r}}}{\partial t} &\approx \frac{\underline{\mathbf{r}}_{t+\Delta t} - \underline{\mathbf{r}}_{t-\Delta t}}{2\Delta t}. \end{aligned} \tag{16}$$

By inserting these derivatives into (15) we convert the nonlinear system into the system of linear equations

$$\mathbf{A}_t \underline{\mathbf{r}}_{t+\Delta t} = \underline{\mathbf{g}}_t \quad \Leftrightarrow \quad \underline{\mathbf{r}}_{t+\Delta t} = \mathbf{A}_t^{-1}\underline{\mathbf{g}}_t\,, \tag{17}$$

where $\mathbf{A}$ is called the system matrix,

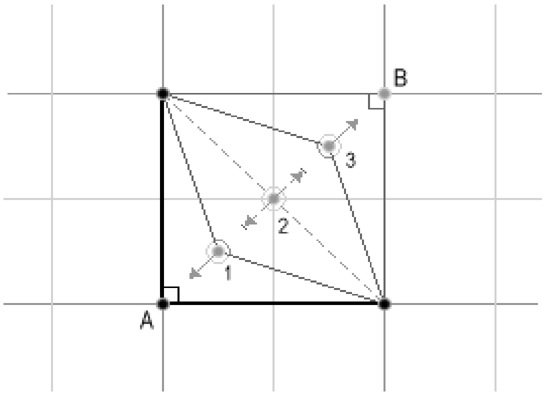

Figure 2 The surface patch will collapse to a curve, when a particle crosses the opposite diagonal

$$\mathbf{A}_t = \mathbf{K}(\underline{\mathbf{r}}_t) + \left(\frac{1}{\Delta t^2}\mathbf{M} + \frac{1}{2\Delta t}\mathbf{C}\right), \tag{18}$$

and $\mathbf{g}$ is called the effective force vector,

$$\underline{\mathbf{g}}_t = \underline{\mathbf{f}}_t + \left(\frac{1}{\Delta t^2}\mathbf{M} + \frac{1}{2\Delta t}\mathbf{C}\right)\underline{\mathbf{r}}_t + \left(\frac{1}{\Delta t}\mathbf{M} - \frac{1}{2}\mathbf{C}\right)\underline{\mathbf{v}}_t\,, \tag{19}$$

with the velocity estimated by the normal first order backwards difference

$$\underline{\mathbf{v}}_t = \Delta t^{-1}\left(\underline{\mathbf{r}}_t - \underline{\mathbf{r}}_{t-\Delta t}\right). \tag{20}$$

With these equations in hand it is possible to implement real time dynamic simulations of deformable solids. The desirable properties of the system matrix indicate that a specialized solution method, such as the conjugate gradient method [8], can be utilized.

## 3 Instabilities

The instabilities we will address are not numerical in nature, such as the chosen integration method, timesteps, etc. We will turn the interest towards instabilities, regarding the structure of the underlying constraints in the model of elastic solids. The problems can, more or less, be identified as boundary issues, as the integrity instabilities arise from the boundaries of the discrete 3D grid. Real life deformable objects are held together by strong forces at a very small scale. Similarly, we could solve the integrity problems, if we could get away with using extremely high values in the underlying weighting tensors. The reason why we must disregard this option is because it will require us to integrate numerically using “infinitely” small timesteps. As we are interested in interactive simulations, we seek to use large timesteps, thus to simply increase the values of the tensors is not a desired option.

Differential Geometry [7] is used as a tool to measure deformation of an elastic body, in comparison with its resting shape. For solids, the first fundamental forms, or $3 \times 3$ metric tensors, are sufficient to distinguish between the shapes of two bodies. However, the metric tensor of a solid (2) is not nearly sufficient enough as a tool to compute the complex particle movements of a deformed solid, seeking towards its resting shape. The discrete components of (2), disregarding the diagonal, describe the cosine to the angle between adjacent directions multiplied by the product of the lengths,

$$\mathbf{v} \cdot \mathbf{w} = |\mathbf{v}||\mathbf{w}| \cos\theta, \quad 0 \leq \theta \leq \pi \,. \tag{21}$$

The angle between two vectors is not dependent on their mutual orientation, as verified by the domain of $\theta$ in (21). This is the main reason why we will get internal structure instabilities. The case of surface patches is depicted in Figure 2. The bold lines and the angle between them form the natural condition. If particle $A$ is moved towards $B$, as in case 1, the angular elasticity constraints will force the particle back to its resting condition, in the direction indicated by the arrow. The behavior of the angular constraint force is defined ambiguously when the angle between vectors is either $0$ or $180$ degrees. The latter is shown in case 2. If the particle reaches beyond the opposite diagonal, as in case 3, the elasticity will be strengthened and push particle $A$ into $B$. This is clearly a problem, as it will reduce the surface into a curve. The original model [9] suffers from this instability.

The instabilities for solids get even worse. Expanding the square to a cube and focusing on a particle in the grid. Not only does the particle spawn surface instabilities for the 3 directional patches, there is also nothing that prevents the cube from collapsing over the space diagonals. The disability of volume preservation will ruin the integrity of the solid.

In short, the original elastic model is insufficient to simulate deformable solids. Based on the modeling concept by Terzopoulos et al [9], we will remodel the method, to render it more suitable for simulating elastically deformable solids.

## 4 Integrity Improvements

To handle integrity instabilities, we are going to design an extension to the original model that can improve its ability to prevent collapsing of the grid cubes. Basically, the extension will be done by adding new constraints to the model, gathered into a new tensor we call the Spatial Diagonal Metric, or SDM. To make things a little easier, we start out by describing the analogy for surfaces.

### 4.1 Extended Metric

For a surface, the original tension constraints on a given particle is the four constraints given by its $2 \times 2$ metric tensor $\mathbf{G}$. As with solids, when $i = j$ then $G_{ij}$ defines the length constraints along the coordinate directions $m$ and $n$, while when $i \neq j$ then $G_{ij}$ represents an expression of the same explicit angular constraint between the two directions. The idea is to replace the dual explicit angular constraint with two diagonal length constraints. These constraints will reach from the particle at $[m,n]$ to the diagonally opposite particles at $[m+1,n+1]$ and $[m-1,n+1]$, as depicted in Figure 3. Besides working as diagonal length constraints, they will also implicitly work as angular constraints that together can account for all $360$ degrees. The directions along these new constraints will be considered as new directions called $a_{d1}$ and $a_{d2}$. When the model is extended with the new diagonal constraints, variational calculus must be utilized for the expression of the minimizing elastic force. Writing out (5) for the case of surfaces, with focus on the new directions, yields

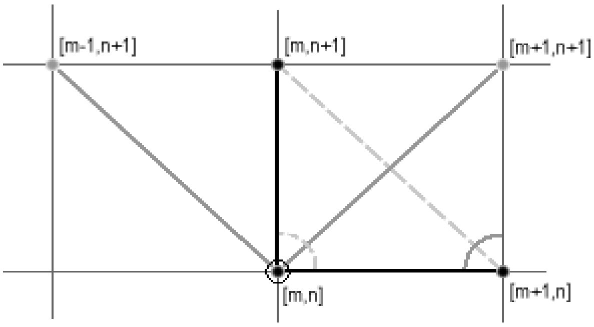

Figure 3 The dual explicit angular constraint is replaced by two new diagonal constraints that will define the angular constraints implicitly

$$\begin{aligned} S = \cdots &+ \eta_{12}\left(\mathbf{r}_{a_{d1}} \cdot \mathbf{r}_{a_{d1}} - \mathbf{G}^0_{12}\right)^2 \\ &+ \eta_{21}\left(\mathbf{r}_{a_{d2}} \cdot \mathbf{r}_{a_{d2}} - \mathbf{G}^0_{21}\right)^2 + \cdots, \end{aligned} \tag{22}$$

where the elements $G^0_{12}$ and $G^0_{21}$ of the rest state tensor, now holds the natural state of the new diagonal constraints. With the new directions, $a_{d1}$ and $a_{d2}$, in hand, the Euler-Lagrange equation changes to

$$\frac{\delta S}{\delta \mathbf{r}} = S_{\mathbf{r}} - \partial_{a_1} S_{\mathbf{r}_{a_1}} - \partial_{a_2} S_{\mathbf{r}_{a_2}} - \partial_{a_{d1}} S_{\mathbf{r}_{a_{d1}}} - \partial_{a_{d2}} S_{\mathbf{r}_{a_{d2}}} \,. \tag{23}$$

Using variational calculus on the terms from (22) results in the following additions to the elastic force $\mathbf{e}$,

$$\mathbf{e}[m,n] = \cdots - D^-_{d1}(\mathbf{p}^*)[m,n] - D^-_{d2}(\mathbf{q}^*)[m,n], \tag{24}$$

where

$$\mathbf{p}^*[m,n] = \eta_{12}[m,n] D^+_{d1}(\mathbf{r})[m,n]$$

and

$$\mathbf{q}^*[m,n] = \eta_{21}[m,n] D^+_{d2}(\mathbf{r})[m,n]. \tag{25}$$

Notice that new difference operators arise with the new directions in the discretization of the extended metric. These operators can be considered to be just like the operators in the original directions. For example, the new first order forward difference operators become

$$\begin{aligned} D^+_{d1}(\mathbf{u}) &= \frac{\mathbf{u}[m+1,n+1] - \mathbf{u}[m,n]}{h_{d1}} \\ D^+_{d2}(\mathbf{u}) &= \frac{\mathbf{u}[m-1,n+1] - \mathbf{u}[m,n]}{h_{d2}}, \end{aligned} \tag{26}$$

where $h_{d1} = h_{d2} = \sqrt{h_1^2 + h_2^2}$ is the grid distance in both diagonal directions. The square root is computationally expensive, but it can easily be avoided in any calculation involving the new difference operators, or simply be precomputed as the value does not change at runtime.

We have shown how to extend the metric tensor, and we call the extended metric for Metrix. To give a deformable solid a stronger integrity, the Metrix can easily be applied to work for solids. The $3 \times 3$ metric tensor for a solid contains three length constraints along its diagonal. The remaining six elements represent angular constraints between the original coordinate directions, and these can be replaced by the appropriate elements of the Metrix.

## 4.2 Spatial Diagonal Metric

To implement volume preservation, we introduce another new idea called the Spatial Diagonal Metric, which is a $2 \times 2$ tensor. The four elements will represent four new length constraints that will be spatially diagonal, meaning they will span grid cubes volumetrically. The four new directions can be chosen three-ways, and we have simply chosen them to reach from $[l, m, n]$ to the four particles at $[l-1, m+1, n+1]$, $[l-1, m-1, n+1]$, $[l+1, m+1, n+1]$, and $[l+1, m-1, n+1]$. No matter how the four length constraints are chosen to span the cubes, their contributions will end up covering a grid cube symmetrically, as depicted in Figure 4.

The new SDM tensor that represents the volume will be called $\mathbf{V}$. It follows the same design ideas as the Metrix, and is defined as

$$\mathbf{V} = \begin{bmatrix} D_{v1}^{+} \cdot D_{v1}^{+} & D_{v2}^{+} \cdot D_{v2}^{+} \\ D_{v3}^{+} \cdot D_{v3}^{+} & D_{v4}^{+} \cdot D_{v4}^{+} \end{bmatrix}, \tag{27}$$

where $D_{v1..4}^{+}$ are the four new first order forward difference operators along the new spatial directions.

# 5 Global Implosions

With the SDM and Metrix contributions added to the elasticity term, the model of deformable solid can prevent the shape of the grid cubes from undergoing local collapsing. This is an important improvement towards keeping the integrity of a deformable solid intact. Another integrity issue still exists that renders a solid unable to prevent implosions. In this matter, we define an implosion as when a grid cube enters one of its adjacent grid cubes. Implosions can happen upon large deformation, which typically are caused by heavy external forces, e.g. reaction to collisions and aggressive user interactions. Global implosions can also be defined as internal self-intersections, thus self-intersection tests can be utilized as a tool to prevent implosions.

Common methods to avoid self-intersection include surrounding each grid cube into an axis aligned bounding box, or AABB, and arrange the AABBs into a bounding volume hierarchy, or BVH tree, that can be updated efficiently as the body deforms [4]. A recent paper introduces image-space techniques that can be implemented on the GPU, to allow performance friendly detection of self-intersections and collision between deformable bodies [6]. Although a method for handling self-intersection can be chosen to perform reasonable, it will still decrease the overall performance.

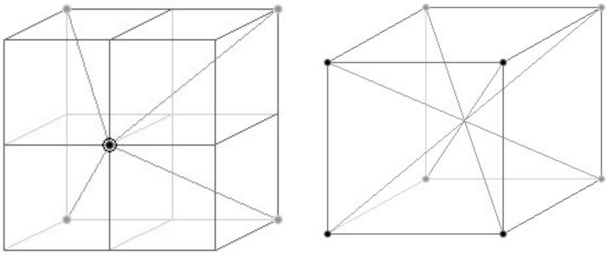

**Figure 4** Spatial length constraints for solids. On the left it is shown how the four constraints reach from the center. On the right it is shown how the constraint contribution from four particles on one cube side renders symmetric behavior

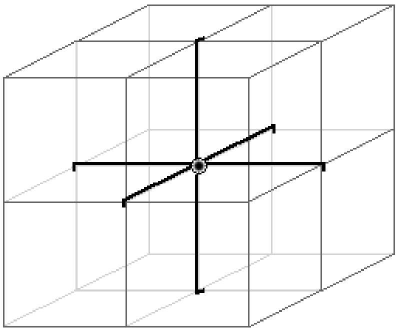

**Figure 5** Discrete central differences bind adjacent grid cubes together

What we seek is a mechanism that somehow binds adjacent grid cubes together, in such a way that if implosions occur, we can disperse self-intersecting cubes. This is not a method that can prevent self-intersections, but it can restore the integrity of the solid upon implosions. We can reuse what we have been working with so far, and thus reduce the computational cost and memory use significantly, compared to the extra load we would introduce into the system, if we had used a BVH algorithm.

We introduce a new Pillar tensor $\mathbf{P}$, which is based upon the discrete metric tensor $\mathbf{G}$, but extended to use first order central difference operators. For reasons of clarity we will limit $\mathbf{P}$ only to use the length constraints

$$\mathbf{P}[l, m, n] = \begin{bmatrix} D_1^2(\mathbf{r}) & 0 & 0 \\ 0 & D_2^2(\mathbf{r}) & 0 \\ 0 & 0 & D_3^2(\mathbf{r}) \end{bmatrix}, \tag{28}$$

where

$$\begin{aligned} D_1(u)[l,m,n] &= ½h_1^{-1}\left(u(l+1,m,n) - u(l-1,m,n)\right), \\ D_2(u)[l,m,n] &= ½h_2^{-1}\left(u(l,m+1,n) - u(l,m-1,n)\right), \\ D_3(u)[l,m,n] &= ½h_3^{-1}\left(u(l,m,n+1) - u(l,m,n-1)\right). \end{aligned} \tag{29}$$

The effect of using central difference operators results in a very convincing way to bind adjacent grid cubes together, see

Figure 5. As every grid particle will be extended with the Pillar tensor, the combined range of $\mathbf{P}$ will overlap all grid cubes.

To further strengthen the prevention of implosion, the Pillar tensor can easily be extended to use a central difference Metrix tensor.

## 6 Results

We have extended the previous implementation of the elastically deformable models [1] to support both the Metrix, SDM, and Pillar contributions, when simulating deformable solids. The implementation can be found in [3]. The deformable solids respond naturally to external forces, e.g. gravity and viscosity. The user can interact with the solids, such as constraining particles to a fixed position and performing pulling operations using spring forces. We have also implemented a scaling mechanism, which allows a user to control the overall uniform strength of the tensors. Adjusting the strength scaling interactively can vary the stiffness of a deformable solid in real time. For example simulating the effect of a solid being inflated.

Experiments have revealed that the effects of the Metrix and SDM do not always succeed satisfactorily, in moving particles back to their natural location. In some situations new energy equilibriums arise unnaturally. With the help of the supported visual debugger we have realized that different constraints can work against each other, and the result is that the sum of constraint contributions is zero. To counteract this problem, we have squared the force of the constraints in the SDM (and the Metrix), to make sure they prevail.

To give a reasonable review of how the improved model solves the presented integrity instabilities, we will perform visual comparisons between the original model and the new improved one. In Figure 6, still frames from a small box that is influenced by gravity and collides with a plane, are compared frame to frame between the models. Due to the lack of volume preservation, the constraints of the original model simply cannot keep the shape of the cube. In Figure 7, we have recreated a configuration from [1], comparing two rubber balls with different particle mass. The picture on the left is taken from [1], where the metrics fail to maintain the integrity of the right ball, thus the ball collapses on itself. The picture on the right is simulated using the same parameters but with the improved model, and the integrity of the ball is now strong enough to stay solid. In Figure 8, a test of how well the models can recover from a sudden large deformation is performed.

The improved model enables simulations of situations that are impossible with the original model. In Figure 1, a soft solid is depicted. The solid has been constrained to the ground, and in three of the top corners. Pulling the last corner downwards results in a large deformation and renders convincing material buckling. In Figure 9, the true strength of the Pillar tensor is illustrated, showing an effect of inflation. In Figure 10, some pudding is constrained to the ground, and being twisted by its top face. The sides of the deformable cube skew as is expected of a soft body like pudding. In Figure 11, a large water lily is deformed upon resting on two pearls. The improved model performs a great job in keeping the water lily fluffy.

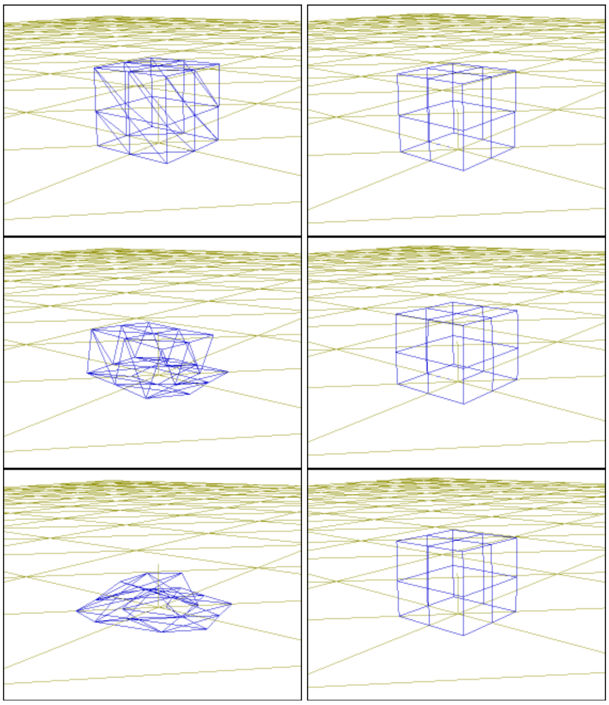

**Figure 6** A small box is influenced by gravity and collides with a plane. The three stills on the left illustrate the original model, and on the right the frames from the new improved model is shown

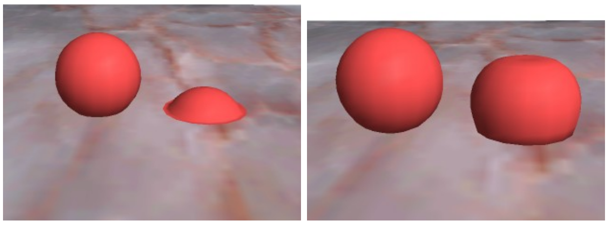

**Figure 7** Rubber balls. The left frame illustrates the situation from the old model where the right ball is unable to maintain its integrity. In the right frame the same situation is depicted, but simulated using the improved model

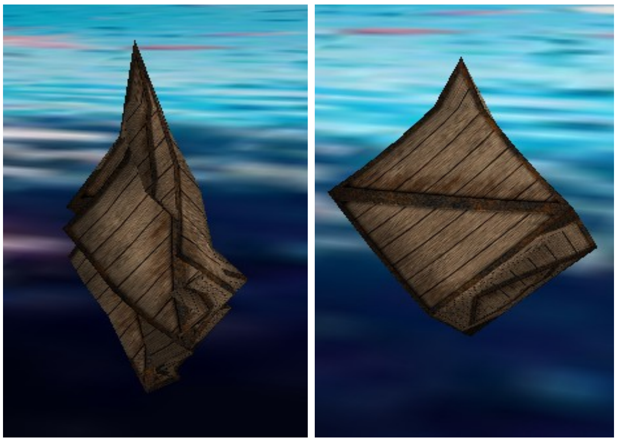

**Figure 8** A wooden box is heavily lifted in one corner. Original vs. improved model, on the left and right frame, respectively

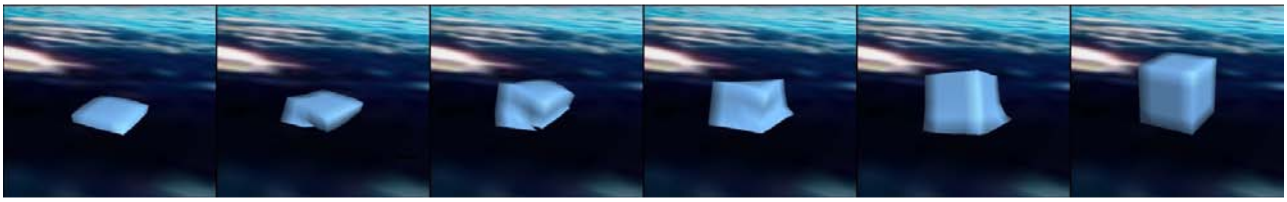

Figure 9 Constraint strength is increased interactively and yields the effect of inflation

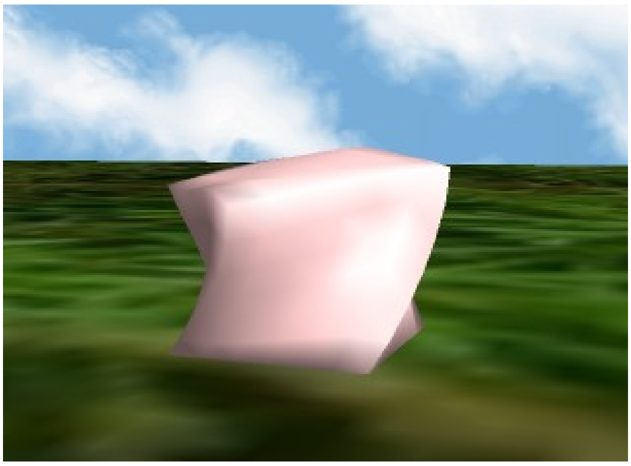

Figure 10 Twisting the pudding renders skewing

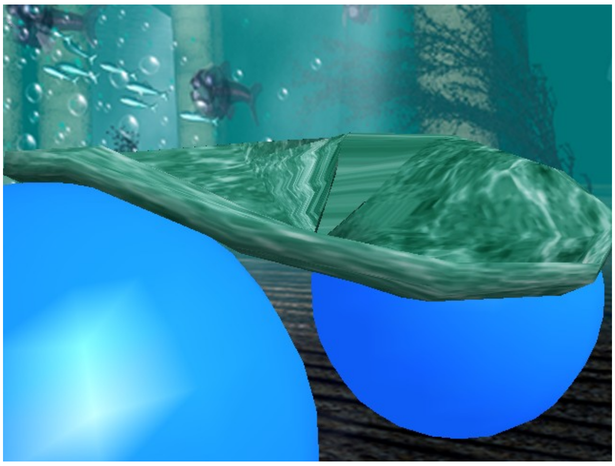

Figure 11 Fluffy water lily modeled using an ellipsoid solid

## 7 Conclusion

The original model for deformable solids, presented in [9], turned out to be insufficient for the sake of realism. Even extremely modest external forces applied to the bodies would ruin their integrity, as can be verified on the left side of Figures 6 to 8. Clearly some extensions to the model were needed. In this paper we have presented three improvements to the model that will give deformable solids a much better ability to keep their original integrity, and thus the ability to handle much larger deformations without collapsing. With the new Metrix, SDM, and Pillar additions to the model, the overall strength of the internal elastic constraints is increased. This reinforces the impression that the elastic bodies are actually solids.

We have shown that the improvements to the original model greatly increase the usability of the method for simulating deformable solids. As the new contributions influence the same particles as the original model, the system matrix $\mathbf{A}$ is still of size $LMN \times LMN$, and it still possess its original pleasant properties that allows an relaxation based solver to invert it, using only a few iterations. The Metrix replaces the metric tensor, thus it is only the SDM and Pillar additions that are new to the calculations, and they can be computed in constant time for each particle.

Newer papers on deformable solids state the importance of preservations of length, surface area, and volume. What we have done with the elastically deformable model is precisely to strengthen the surface preservation using the Metrix, and to implement the missing volume preservation using the SDM.

## References

[1] M. K. Christensen and A. Fleron (2004), *“Implementation of Deformable Objects,”* Department of Computer Science, University of Copenhagen, DIKU

[2] D. Eberly (2003), "*Derivative Approximation by Finite Differences,*" Magic Software, Inc., January 21, 2003

[3] K. Erleben, H. Dohlmann, J. Sporring, and K. Henriksen (2003) *“The OpenTissue project,”* Department of Computer Science, University of Copenhagen, DIKU, November 2003, http://www.diku.dk/forskning/image/research/opentissue/

[4] K. Erleben, J. Sporring, K. Henriksen, and H. Dohlmann (2004), "*Physics-based Animation and Simulation,*" DIKU

[5] H. Goldstein, C. P. Poole, and J. L. Safko (2002), "*Classical Mechanics,*" Third Edition, Addision-Wesley

[6] B. Heidelberger, M. Teschner, and M. Gross (2004), "*Detection of Collisions and Self-collisions Using Image-space Techniques*," Proc. WSCG'04, University of West Bohemia, Czech Republic, pp. 145-152

[7] J. J. Koenderink (1990), "*Solid Shape*," MIT Press

[8] J. R. Shewchuk (1994), "*An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*," Carnegie Mellon University

[9] D. Terzopoulos, J. C. Platt, A. H. Barr, and K. Fleischer (1987), *“Elastically deformable models,”* Computer Graphics, volume 21, Number 4, July 1987, pp 205-214