

# Position-Based Simulation of Continuous Materials

Jan Bender<sup>a</sup>, Dan Koschier<sup>a</sup>, Patrick Charrier<sup>a</sup>, Daniel Weber<sup>b</sup>

<sup>a</sup>Graduate School CE, TU Darmstadt

<sup>b</sup>Fraunhofer IGD, Darmstadt

---

## Abstract

We introduce a novel fast and robust simulation method for deformable solids that supports complex physical effects like lateral contraction, anisotropy or elastoplasticity. Our method uses a continuum-based formulation to compute strain and bending energies for two- and three-dimensional bodies. In contrast to previous work, we do not determine forces to reduce these potential energies, instead we use a position-based approach. This combination of a continuum-based formulation with a position-based method enables us to keep the simulation algorithm stable, fast and controllable while providing the ability to simulate complex physical phenomena lacking in former position-based approaches. We demonstrate how to simulate cloth and volumetric bodies with lateral contraction, bending, plasticity as well as anisotropy and proof robustness even in case of degenerate or inverted elements. Due to the continuous material model of our method further physical phenomena like fracture or viscoelasticity can be easily implemented using already existing approaches. Furthermore, a combination with other geometrically motivated methods is possible.

*Keywords:* position-based simulation, continuum mechanics, finite elements, deformable solids

---

## 1. Introduction

The physically-based simulation of deformable solids is a topic of active research in the field of computer graphics for more than two decades. In general, simulation methods should reflect the behavior of real materials accurately to achieve realistic results. For this reason continuum mechanical approaches in combination with finite element methods (FEM) are widely used. These methods rely on either implicit time integration schemes or very small time steps to provide a robust simulation. Besides the accurate material behavior, many real-time applications focus on robustness and interactivity. Therefore, geometrically motivated approaches were developed to generate stable simulations while keeping computation times low. These methods result in a physically plausible behavior of the solid, but are not able to model complex material properties. Due to the robustness of the methods, they are often used to improve traditional approaches, e.g. strain-limiting methods for cloth simulations.

In this paper we present a novel fast and robust method for the simulation of two- and three-dimensional solids that supports complex physical phenomena. Our approach combines continuum mechanical material models with a position-based energy reduction. Its position-based nature allows to perform a stable simulation using an explicit time integration scheme. Our method also strongly benefits from the continuous model since in contrast to previous position-based approaches it can handle complex physical effects like isotropic and anisotropic elastic behavior as well as the effects of lateral contraction. Moreover, we show how the problem of element inversion can be handled by

only a small modification of the algorithm and provide a formulation to easily embed elastoplastic effects into the simulation. Finally, we demonstrate that our method is able to simulate thousands of degrees of freedom at interactive rates which makes it well-suited for applications like computer games, special effects in movies and virtual reality.

## 2. Related Work

In the last two decades the simulation of deformable solids has been a topic of active research. Nealen et al. [1] presented a general survey of simulation methods. The first physically-based approaches to simulate deformable objects used continuum mechanical formulations, where the governing equations were discretized and solved using numerical integration. Terzopolous et al. [2] used a finite difference scheme to discretize their model based on a non-linear strain measure. Later, O'Brien et al. presented a finite element method for spatial discretization to model brittle [3] and ductile [4] fracture, where explicit time integration schemes were used. However, explicit schemes suffer from stability issues regarding stiff differential equations which forces the usage of very small time steps resulting in high computational costs. James and Pai [5] discretized the equations using a boundary element method and solved the system in a quasi-static context avoiding the usage of a time integration scheme. Unfortunately, dynamic effects are completely neglected making the model unsuitable for scenes with spatially unconstrained, freely moving objects. Since explicit integration methods used in

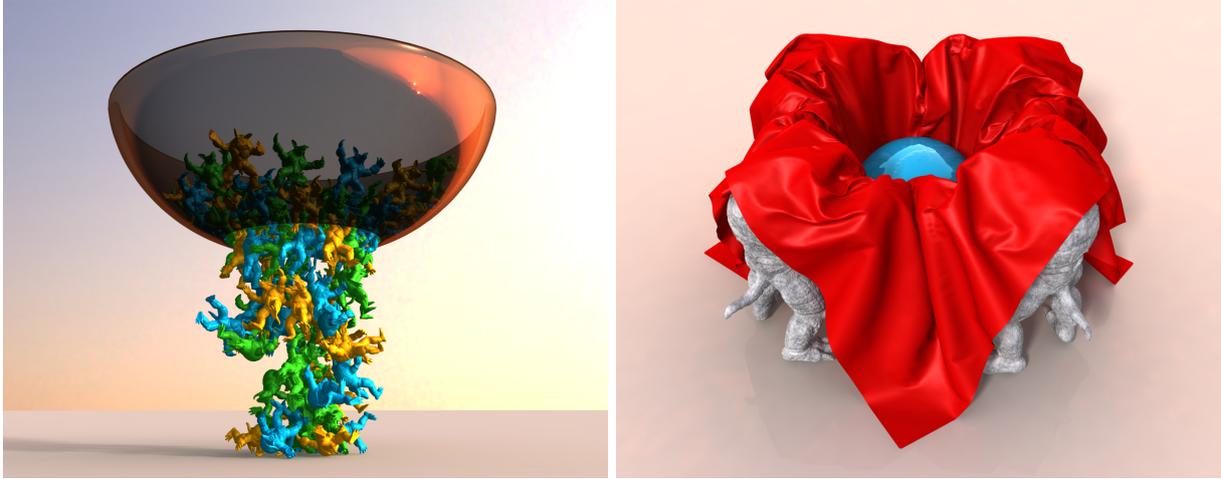


Figure 1: Complex simulations performed with our novel approach. Left: Simulation of 100 Stanford Armadillos with 371700 tetrahedra. Right: Heavy sphere pushing down an orthotropic cloth model that is thrown over four statues. Realistic wrinkles evolve due to the bending resistance.

early works suffer from instabilities, implicit time integration schemes became popular. These schemes are computationally more expensive than explicit ones as large, generally non-linear systems of equations have to be solved. However, they allow to use larger time step sizes. Regarding the finite element method with linear Lagrangian shape functions, the underlying strain measure has to be linearized to keep the systems linear. However, a linearized strain leads to artifacts when deformable bodies are exposed to rotational motion. Eitzmuß et al. [6] introduced a corotational method, where the linearized strain is measured in a local, non-rotated reference frame, obtained by computing a polar decomposition of the deformation gradient. Due to derivatives of the rotational parts this still yields a non-linear system of equations, but nevertheless they found the simulation to be sufficient by neglecting non-linear terms in force derivatives and performing only a single step of Newton’s method. To efficiently solve the linear system of equations multigrid solvers have been employed by Georgii et al. [7] as well as Dick et al. [8], where a hierarchy of discretizations is used to improve the rate of convergence for either short- and long-wavelength components. At the same time GPU-based solvers [8, 9] and adaptive meshes [10] were investigated to simulate complex models in real-time. Another approach to speed up simulation time was made by Hecht et al. [11]. They solved the linear equation system using a sparse Cholesky factorization which is incrementally updated during the simulation yielding a very efficient computation of the solution. Furthermore, the corotated approach was generalized to a discontinuous Galerkin finite element method by Kaufmann et al. [12] to overcome the restrictions of conforming bases. This easily allows the usage of convex as well as non-convex polyhedral elements with simple polynomial shape functions. As previously mentioned the corotational approach ignores higher order derivatives in the stiffness matrix which finally leads to instabilities with large de-

formations. Therefore, McAdams et al. [13] solved the instabilities by proposing a new integration rule for hexahedral elements. Further, Georgii et al. [7] presented a new method to extract the rigid body motion using an energy minimization resulting in a more stable corotational formulation. When considering the higher order terms, the instabilities are minimized but at the cost of solving non-linear systems [14].

Similar to existing continuum-mechanical approaches we use a spatial finite element discretization. A vast majority of simulation methods employ a corotational approach. In contrast to that our method easily allows to use arbitrary non-linear material models that do not rely on a polar decomposition. Moreover, our approach is based on a position-based energy reduction step which allows a stable simulation with an explicit integration scheme.

In the last years position-based methods became popular since they are fast, robust and controllable while no implicit time integration is required. A general survey of position-based methods is presented by Bender et al. [15]. The geometrically motivated, meshless concept of shape matching was introduced by Müller et al. [16] to simulate deformable objects. Traditional forces are avoided in favor of position displacements to solve a geometric constraint. The goal positions are determined by minimizing the distance between the reference shape and the deformed shape of a body. This minimization process requires the computation of translational vectors for both shapes and of a rotation matrix by a polar decomposition. To enhance the efficiency of the method Rivers et al. [17] proposed a fast summation method on regular lattices. Later, Diziol et al. [18] generalized the fast summation approach for irregular structures and introduced a method to enforce incompressibility. Bender et al. [19] introduced a multi-resolution approach to enhance the convergence. Further, Müller et al. [20] added an orientation to each particle to increase the robustness of shape matching. Another ge-

ometrically motivated technique is the position-based dynamics approach introduced by Müller et al. [21]. The authors demonstrated how to simulate cloth models by iteratively solving geometric constraints. Later, the position-based method was extended in order to simulate fluids [22], rigid bodies [23] and elastic rods [24]. Stam [25] also used constraints to simulate the deformation of a solid body, where the main difference to the approach of Müller et al. is that the solver of Stam is velocity-based.

The approach presented in this work integrates seamlessly into the position-based framework of Müller et al., but uses a material model derived from continuum mechanics to enforce physical phenomena that previous formulations are lacking and is even able to resolve degenerate or inverted shapes.

The simulation of cloth is closely related to the simulation of two-dimensional solids except of additional out-of-plane forces that prevent the object from excessive bending. For an overview we refer to the survey of Magnenat-Thalmann and Volino [26]. Early approaches were based on mass-spring systems, e.g. [27]. However, the behavior is difficult to control since the spring stiffnesses are generally unknown for specific materials. Later, also finite element methods solving partial differential equations derived from continuum mechanics were applied to model cloth [6, 28]. Specific material behavior can then be enforced easily since the model provides established parameter sets used in classical mechanics. Müller et al. [21, 29] presented a position-based method for robust interactive simulations. An additional bending constraint model for this position-based approach was introduced by Kelager et al. [30]. Since pieces of cloth only accommodate small amounts of stretching, strain limiting methods were developed. Thomaszewski et al. [31] limit the maximal strain of generally biphasic, anisotropic materials based on a continuum mechanical deformation measure. Later, Wang et al. [32] proposed an isotropic strain limiting where the underlying deformation gradient is directly modified by using a singular value decomposition.

In our approach we combine the advantages of both the continuum mechanical and the position-based approaches to maintain complex physical phenomena while keeping the simulation easy to implement, fast, robust and controllable.

### 3. Position-Based Energy Reduction

In our work we simulate the elasticity of the bodies by reducing potential energy functions  $E(\mathbf{x})$  that correspond to their deformation. This energy reduction is performed using a position-based approach [21]. The concept of this approach is introduced in the following while the required energy functions are presented in Section 4.

#### 3.1. Overview

We use particle meshes to represent deformable bodies in our simulation. Each particle has a mass  $m$ , a position

---

#### Algorithm 1 Simulation step

---

```

1:  $\mathbf{v}^{n+1} \leftarrow \mathbf{v}^n + \Delta t \mathbf{M}^{-1} \mathbf{f}_{\text{ext}}^n$ 
2:  $\mathbf{x}^{n+1} \leftarrow \mathbf{x}^n + \Delta t \mathbf{v}^{n+1}$ 
3: for iter := 1 to maxIterations do
4:   for each energy constraint  $C(\mathbf{x})$  do
5:     compute a Lagrange multiplier  $\lambda$  (Equation (3))
6:     determine position changes  $\Delta \mathbf{x}$  (Equation (2))
7:      $\mathbf{x}^{n+1} \leftarrow \mathbf{x}^{n+1} + \Delta \mathbf{x}$ 
8:   end for
9: end for
10:  $\mathbf{v}^{n+1} \leftarrow \frac{1}{\Delta t} (\mathbf{x}^{n+1} - \mathbf{x}^n)$ 
11: perform continuous collision detection and response
12: damp velocities

```

---

$\mathbf{x}$  and a velocity  $\mathbf{v}$ . Mass lumping is used to concentrate the mass of the model at the vertices of the mesh which yields a diagonal mass matrix  $\mathbf{M}$ .

Position-based simulation methods typically work in three steps [15]. First, a time integration step is performed for a particle model in order to obtain new locations of the particles. These locations are used as predicted positions which are modified in the second step in order to fulfill given position constraints. The definition of bilateral constraints yields a system of equations which is generally non-linear. Solving this system exactly would result in a completely stiff body. Therefore, Müller et al. [21] propose to use an iterative solver and to perform only a few iterations which yields an elastic behavior and allows a high performance. In the final step the corrected positions are used to update the velocities of the particles.

Algorithm 1 outlines a simulation step with our position-based energy reduction method. In the lines (1) and (2) the current velocities  $\mathbf{v}^n$  and positions  $\mathbf{x}^n$  are integrated using a symplectic Euler scheme, where  $\Delta t$  is the time step size and  $\mathbf{f}_{\text{ext}}^n$  are external forces. Note that the symplectic Euler method differs only slightly from the explicit Euler scheme and has the same computational effort but better properties concerning stability and energy conservation [33]. In the time integration step only external forces are considered which corresponds to an integration of unconstrained particles. The resulting positions  $\mathbf{x}^{n+1}$  are modified by the iterative energy reduction process in lines (3)-(9) to simulate the elastic behavior of our bodies. This corresponds to a static solver which corrects the positions in order to meet given constraints. The modified positions are used to update the velocities in line (10). Finally, collision handling is performed and the velocities are damped. In our work we use the collision handling proposed by Bridson et al. [34].

#### 3.2. Position-Based Solver

After computing the predicted positions in line (2) of Algorithm 1, these positions are modified to fulfill a set of bilateral position constraints  $C(\mathbf{x}) = 0$ . The goal of the position-based solver is to find a position corrections  $\Delta \mathbf{x}$

so that  $C(\mathbf{x} + \Delta\mathbf{x}) = 0$ . In general the resulting system of equations is non-linear. It can be solved using the Newton-Raphson method which iteratively solves a linearized system of equations. However, since this is computationally expensive, we use a modified version of the Newton-Raphson solver. Each constraint is linearized and solved independently while the solver iterates multiple times over all constraints in a Gauss-Seidel fashion. For a single constraint the position correction  $\Delta\mathbf{x}$  is computed by solving the linearized equation

$$C(\mathbf{x} + \Delta\mathbf{x}) \approx C(\mathbf{x}) + \nabla_{\mathbf{x}} C^T(\mathbf{x})\Delta\mathbf{x} = 0. \quad (1)$$

The direction of the position correction is restricted to solve this underdetermined equation. By performing the position correction only in the direction of  $\nabla_{\mathbf{x}} C(\mathbf{x})$  the linear and angular momenta are conserved which corresponds to the principle of virtual work [35]. Considering also the individual masses of the particles, the position change of a particle  $i$  is determined by

$$\Delta\mathbf{x}_i = w_i \lambda \nabla_{\mathbf{x}_i} C(\mathbf{x}), \quad (2)$$

where  $\lambda$  is a Lagrange multiplier and  $w_i = 1/m_i$  is the inverse particle mass. Substituting this position change in Equation (1) yields

$$\lambda = -\frac{C(\mathbf{x})}{\sum_j w_j |\nabla_{\mathbf{x}_j} C(\mathbf{x})|^2}. \quad (3)$$

Since the constraints are linearized independently one after another, each constraint function  $C(\mathbf{x})$  and its corresponding gradient  $\nabla_{\mathbf{x}} C(\mathbf{x})$  already consider the positions changes of previous constraints. This yields a good convergence as shown in Section 6. In contrast the Newton-Raphson method evaluates all values at the beginning of an iteration step and solves the linearized system without adapting them. For this method a model with  $n$  constraints requires to solve a  $n \times n$  system of linear equations in each iteration step while the modified Newton-Raphson method proposed above only solves  $n$  independent linear equations per step. Therefore, the modified version is significantly faster than the original Newton-Raphson method while the convergence of the latter one is only slightly better which is shown in Section 6.

A simple example of a position constraint is the distance constraint  $C(\mathbf{x}_1, \mathbf{x}_2) = |\mathbf{x}_1 - \mathbf{x}_2| - d_0$ , where  $d_0$  is the rest distance between the constrained particles. This constraint was used by Müller et al. [21] to simulate cloth models. The position corrections for this constraint are determined by Equations (2) and (3):

$$\begin{aligned} \Delta\mathbf{x}_1 &= -\frac{w_1}{w_1 + w_2} (|\mathbf{x}_1 - \mathbf{x}_2| - d_0) \frac{\mathbf{x}_1 - \mathbf{x}_2}{|\mathbf{x}_1 - \mathbf{x}_2|} \\ \Delta\mathbf{x}_2 &= +\frac{w_1}{w_1 + w_2} (|\mathbf{x}_1 - \mathbf{x}_2| - d_0) \frac{\mathbf{x}_1 - \mathbf{x}_2}{|\mathbf{x}_1 - \mathbf{x}_2|}. \end{aligned}$$

In contrast to previous works which use such geometric constraints to simulate deformable models, we propose

to use energy constraints. In the next section we introduce energy functions  $E(\mathbf{x})$  which are based on continuum mechanics. In order to simulate deformable solids with the position-based approach we define an energy constraint  $C(\mathbf{x}) = E(\mathbf{x}) = 0$  for each energy function. The advantage of using energy functions instead of geometric constraints is that thanks to the continuum-based formulation we use, physical phenomena like lateral contraction, anisotropy or elastoplasticity can be simulated since the characteristic deformation behavior of these effects is encoded in the energy functions and their gradients.

## 4. Deformable Solids

In this section we show how soft bodies and cloth models are simulated with the position-based energy reduction approach. In order to model elastic behavior we define a strain energy function  $E_s$  on the considered domain  $\Omega$  derived from continuum mechanics and reduce the implied energy of the body using the method explained in Section 3. Regarding cloth simulation we use an additional bending energy  $E_b$  as the development of folds and wrinkles is essential for realistic results.

### 4.1. Elastic Energy

The deformation of a body is described by a continuous displacement field  $\mathbf{u}$  [36]. This displacement field is used to define the deformation function

$$\phi(\mathbf{X}) = \mathbf{X} + \mathbf{u} = \mathbf{x}$$

which maps a point  $\mathbf{X}$  in material space to its deformed location  $\mathbf{x}$  in world space. The deformation gradient is defined by the Jacobian of the deformation mapping

$$\mathbf{F} = \frac{\partial \phi(\mathbf{X})}{\partial \mathbf{X}}.$$

This gradient is required to determine the non-linear Green strain tensor

$$\boldsymbol{\epsilon} = \frac{1}{2} (\mathbf{F}^T \mathbf{F} - \mathbf{I}),$$

where  $\mathbf{I}$  denotes the identity matrix.

Our simulation method supports different constitutive material models. We demonstrate this using the Saint Venant-Kirchhoff model and the Neo-Hookean model which are introduced in the following.

*Saint Venant-Kirchhoff Model.* The relation between stress and strain is modeled using Hooke's generalized law

$$\mathbf{S} = \mathbf{C}\boldsymbol{\epsilon}, \quad (4)$$

where  $\mathbf{C}$  is the elasticity tensor of fourth order modeling the elastic behavior of the material. Due to symmetries  $\mathbf{C}$  inherits 21 independent entries for volumetric bodies. For orthotropic materials the number reduces to nine entries for three-dimensional continua. For isotropic materials only two independent entries in  $\mathbf{C}$  remain, which are

often expressed by the engineering constants Young's modulus  $k$  and Poisson ratio  $\nu$ .

For isotropic materials the stress-strain relation in Equation (4) is called Saint Venant-Kirchhoff model. The scalar strain energy density field is then defined by

$$\Psi_s = \frac{1}{2} \boldsymbol{\epsilon} : \mathbf{S} = \frac{1}{2} \boldsymbol{\epsilon} : \mathbf{C} \boldsymbol{\epsilon}$$

using the inner product  $\mathbf{A} : \mathbf{B} = \text{tr}[\mathbf{A}^T \mathbf{B}]$ , where  $\text{tr}(\cdot)$  is the trace of a matrix. Then the resulting energy stored in the solid is

$$E_s = \int_{\Omega} \Psi_s d\mathbf{X}. \quad (5)$$

To reduce this energy the method presented in Section 3 makes use of the world space gradient which can also be interpreted as an inner force field acting from inside the body

$$\nabla_{\mathbf{x}} E_s = \int_{\Omega} \frac{\partial \Psi_s}{\partial \mathbf{x}} d\mathbf{X} = \int_{\Omega} \frac{\partial \Psi_s}{\partial \mathbf{F}} \frac{\partial \mathbf{F}}{\partial \mathbf{x}} d\mathbf{X} = \int_{\Omega} \mathbf{P}(\mathbf{F}) \frac{\partial \mathbf{F}}{\partial \mathbf{x}} d\mathbf{X}, \quad (6)$$

where

$$\mathbf{P}(\mathbf{F}) = \mathbf{F} \mathbf{C} \boldsymbol{\epsilon} \quad (7)$$

is the first Piola-Kirchhoff stress tensor.

*Neo-Hookean Model.* Another isotropic constitutive model is the Neo-Hookean model which is defined by the isotropic invariants:

$$I_1 = \text{tr}(\mathbf{F}^T \mathbf{F}), \quad I_3 = \det(\mathbf{F}^T \mathbf{F}).$$

For this model the strain energy density field is

$$\Psi_s = \frac{\mu}{2} (I_1 - \log(I_3) - 3) + \frac{\lambda}{8} \log^2(I_3),$$

where  $\mu$  and  $\lambda$  are the Lamé coefficients:

$$\mu = \frac{k}{2(1+\nu)}, \quad \lambda = \frac{k\nu}{(1+\nu)(1-2\nu)}.$$

The first Piola-Kirchhoff stress tensor is then defined as

$$\mathbf{P}(\mathbf{F}) = \mu \mathbf{F} - \mu \mathbf{F}^{-T} + \frac{\lambda \log(I_3)}{2} \mathbf{F}^{-T}. \quad (8)$$

#### 4.2. Soft Bodies

*Discretization.* In order to perform a numerical simulation of a deformable body, we discretize the body using tetrahedral elements with linear Lagrangian shape functions as described by Sifakis and Barbic [37]. The deformation gradient for such an element is

$$\mathbf{F} = \mathbf{D}_s \mathbf{D}_m^{-1}, \quad (9)$$

where  $\mathbf{D}_s$  is the deformed shape matrix and  $\mathbf{D}_m$  the constant reference shape matrix defined by

$$\mathbf{D}_s = \begin{pmatrix} \mathbf{x}_1 - \mathbf{x}_4 & \mathbf{x}_2 - \mathbf{x}_4 & \mathbf{x}_3 - \mathbf{x}_4 \end{pmatrix}$$

$$\mathbf{D}_m = \begin{pmatrix} \mathbf{X}_1 - \mathbf{X}_4 & \mathbf{X}_2 - \mathbf{X}_4 & \mathbf{X}_3 - \mathbf{X}_4 \end{pmatrix}.$$

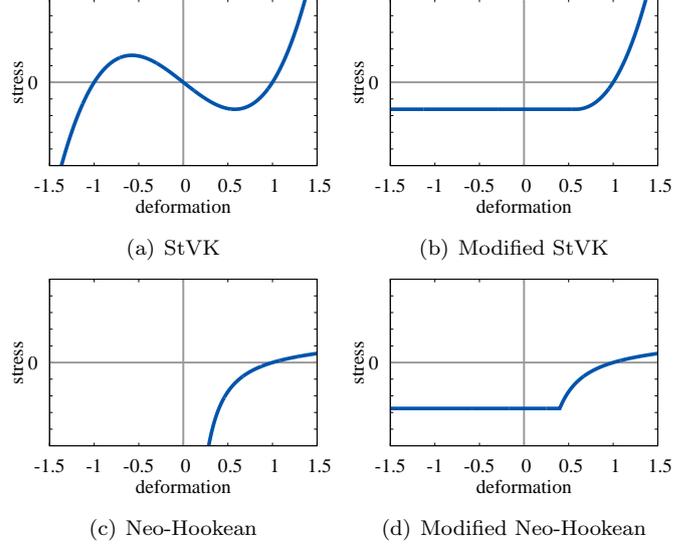


Figure 2: (a,c) The relationship between the first component of the diagonalized first Piola-Kirchhoff stress tensor  $\hat{\mathbf{P}}_{x,x}$  and the corresponding component of the deformation gradient  $\hat{\mathbf{F}}_{x,x}$  for the Saint Venant-Kirchhoff model (StVK) and the Neo-Hookean model, respectively. (b,d) The relationship is modified to handle degenerate and inverted elements robustly.

Since the deformation gradient has only support on the tetrahedral domain  $\Omega_{\Delta} \subseteq \Omega$  and since  $\mathbf{F}$  is constant with respect to material space coordinates  $\mathbf{X}$ , the strain energy stored in a single element is according to Equation (5)

$$E_s = \int_{\Omega_{\Delta}} \Psi_s(\mathbf{F}) d\mathbf{X} = V \Psi_s(\mathbf{F}),$$

where  $V$  is the undeformed volume of the element. This gives us the energy function required for the position-based energy reduction approach. Furthermore, we need to determine the corresponding strain energy gradient  $\partial E_s / \partial \mathbf{x}_i$  for each vertex  $i$  of an element. These gradients are according to Equation (6)

$$\left[ \frac{\partial E_s}{\partial \mathbf{x}_1} \quad \frac{\partial E_s}{\partial \mathbf{x}_2} \quad \frac{\partial E_s}{\partial \mathbf{x}_3} \right] = V \mathbf{P}(\mathbf{F}) \mathbf{D}_m^{-T}, \quad \frac{\partial E_s}{\partial \mathbf{x}_4} = - \sum_{i=1}^3 \frac{\partial E_s}{\partial \mathbf{x}_i}.$$

*Inversion Handling.* When simulating large deformations, the inversion of tetrahedral elements cannot be avoided. Common constitutive models are not designed to handle such configurations. To restore inverted elements robustly we use the approach of Irving et al. [38]. Inverted elements are detected by checking the sign of  $\det \mathbf{F}$ . For such an element a singular value decomposition of  $\mathbf{F} = \mathbf{U} \hat{\mathbf{F}} \mathbf{V}^T$  is computed, where  $\mathbf{U}$  and  $\mathbf{V}$  are rotation matrices. The matrix  $\mathbf{U}$  contains a reflection for inverted tetrahedra which is removed by negating the smallest element of  $\hat{\mathbf{F}}$  and its corresponding column in  $\mathbf{U}$ . The diagonalized first Piola-Kirchhoff stress tensor  $\hat{\mathbf{P}} = \mathbf{P}(\hat{\mathbf{F}})$  is determined by Equation (7) or (8) depending on the constitutive model.

When using a Saint Venant-Kirchhoff material, a compressed elastic body reacts with a restorative force which

reaches a maximum at a compression of approximately 58% (see Figure 2(a)). As suggested by Gao et al. [39], we clamp the stress at its maximum value to handle degenerate and inverted elements robustly. This means that we modify  $\hat{\mathbf{P}}$  according to Figure 2(b). Finally, we obtain the modified stress tensor by  $\mathbf{P} = \mathbf{U}\hat{\mathbf{P}}\mathbf{V}^T$ . The handling of degenerate and inverted elements for other constitutive models can be performed analogously as shown in Figures 2(c) and 2(d) for the Neo-Hookean model.

*Plastic Deformation.* In our work we simulate plastic deformations by a simple and efficient method which was inspired by the work of Müller and Gross [40]. We additionally decompose the total strain  $\boldsymbol{\epsilon}$  into an elastic and a plastic component:

$$\boldsymbol{\epsilon} = \boldsymbol{\epsilon}_{\text{elastic}} + \boldsymbol{\epsilon}_{\text{plastic}}.$$

To update the plastic component, we use the deviation of the elastic strain which is determined by

$$\boldsymbol{\epsilon}' = \boldsymbol{\epsilon}_{\text{elastic}} - \frac{\text{tr}(\boldsymbol{\epsilon}_{\text{elastic}})}{3}\mathbf{I}.$$

As proposed by O'Brien et al. [4], we employ von Mises's yield criterion

$$\gamma_{\text{yield}} < \|\boldsymbol{\epsilon}'\|$$

to determine when plastic flow begins, where  $\|\cdot\|$  is the Frobenius norm and  $\gamma_{\text{yield}}$  a user-defined material parameter. If this criterion is satisfied, plastic deformation will occur and the plastic strain absorbs a portion of  $\boldsymbol{\epsilon}'$ . In this case we update the plastic component of the strain tensor according to:

$$\boldsymbol{\epsilon}_{\text{plastic}} := \begin{cases} \boldsymbol{\epsilon}_{\text{plastic}} + \Delta t \gamma_{\text{creep}} \boldsymbol{\epsilon}' & \text{if } \gamma_{\text{yield}} < \|\boldsymbol{\epsilon}'\| < \gamma_{\text{max}} \\ \gamma_{\text{max}} \frac{\boldsymbol{\epsilon}_{\text{plastic}}}{\|\boldsymbol{\epsilon}_{\text{plastic}}\|} & \text{if } \|\boldsymbol{\epsilon}'\| \geq \gamma_{\text{max}}, \end{cases}$$

where  $\gamma_{\text{creep}}$  is the plastic flow rate and  $\gamma_{\text{max}}$  determines the maximum plastic strain.

As an alternative to the additive strain decomposition a multiplicative decomposition of the deformation gradient could also be used as proposed in [38]. In contrast to the additive model additional constraints such as incompressibility can be enforced easily regarding traditional finite element simulations. On the other hand our additive decomposition is more efficient in terms of computational effort since the plasticity update in combination with the multiplicative model of Irving et al. [38] relies on a singular value decomposition.

### 4.3. Cloth

*Discretization.* We discretize a cloth model using a triangle mesh with linear Lagrangian shape functions. For each triangle we project its deformed configuration and its reference configuration on the plane of the triangle to obtain 2D coordinates. Analogously to Equation (9), the deformation gradient  $\mathbf{F} \in \mathbb{R}^{2 \times 2}$  of a triangle is determined by

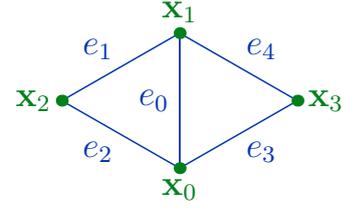


Figure 3: The stencil of an interior edge  $e_0$  which is used to determine the bending energy.

its deformed shape matrix and its reference shape matrix. The strain energy of a triangular element is

$$E_s = A\Psi(\mathbf{F}),$$

where  $A$  is the area of the undeformed triangle. The strain energy gradients of the three vertices are determined by

$$\left[ \frac{\partial E_s}{\partial \mathbf{x}_1} \quad \frac{\partial E_s}{\partial \mathbf{x}_2} \right] = A\mathbf{P}(\mathbf{F})\mathbf{D}_m^{-T}, \quad \frac{\partial E_s}{\partial \mathbf{x}_3} = - \sum_{i=1}^2 \frac{\partial E_s}{\partial \mathbf{x}_i}.$$

*Anisotropy.* Woven fabrics have a weft and a warp direction which leads to an orthotropic behavior which is a special case of anisotropy. We model this behavior by using an orthotropic material with two orthogonal symmetry axes. The corresponding elasticity tensor is defined by two Young's moduli  $k_{\text{weft}}$  and  $k_{\text{warp}}$ , a shear modulus  $k_{\text{shear}}$  and the Poisson ratios  $\nu_{\text{weft}}$  and  $\nu_{\text{warp}}$ :

$$\mathbf{C} = \begin{pmatrix} \frac{k_{\text{weft}}}{1-\nu_{\text{weft}}\nu_{\text{warp}}} & \frac{k_{\text{weft}}\nu_{\text{warp}}}{1-\nu_{\text{weft}}\nu_{\text{warp}}} & 0 \\ \frac{k_{\text{warp}}\nu_{\text{weft}}}{1-\nu_{\text{weft}}\nu_{\text{warp}}} & \frac{k_{\text{warp}}}{1-\nu_{\text{weft}}\nu_{\text{warp}}} & 0 \\ 0 & 0 & k_{\text{shear}} \end{pmatrix}.$$

Note that other anisotropic materials can be simulated as well by using a corresponding elasticity tensor.

*Bending.* We use the discrete isometric bending model of Bergou et al. [41] to simulate cloth with folds and wrinkles. For an interior edge  $e_i$ , we define a stencil  $s$  consisting of the two triangles adjacent to  $e_i$ , where  $\mathbf{x}_s = (\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)^T$  contains the four vertices of the stencil and  $\mathbf{e}_s = [\mathbf{x}_0\mathbf{x}_1, \mathbf{x}_1\mathbf{x}_2, \mathbf{x}_2\mathbf{x}_0, \mathbf{x}_0\mathbf{x}_3, \mathbf{x}_3\mathbf{x}_1]$  is a list of the five stencil edges starting with the common edge (see Figure 3). The local Hessian bending energy of a stencil is obtained as

$$\mathbf{Q}^s = \frac{3}{A_0 + A_1} \mathbf{K}^T \mathbf{K},$$

where  $A_0$  and  $A_1$  are the areas of the adjacent triangles and  $\mathbf{K}$  is the vector

$$\mathbf{K} = (c_{01} + c_{04}, c_{02} + c_{03}, -c_{01} - c_{02}, -c_{03} - c_{04}),$$

where  $c_{jk} = \cot \angle e_j, e_k$ . The bending energy of the stencil is defined as

$$E_b(\mathbf{x}_s) = \frac{1}{2} \sum_{i,j} \mathbf{Q}_{i,j}^s \mathbf{x}_i^T \mathbf{x}_j.$$

Finally, the local bending energy gradients are determined by

$$\frac{\partial E_b}{\partial \mathbf{x}_i} = \sum_j \mathbf{Q}_{i,j}^s \mathbf{x}_j.$$

## 5. Visualization

Since the surface resolution of a tetrahedral model is often too low for fine details, a common visualization method for soft bodies is to embed a high resolution rendering geometry in the tetrahedral mesh. However, this method is limited since the rendering mesh must be fully contained within the volume mesh and since it cannot handle cloth models with details above the surface mesh. In our work we use the approach of Kobbelt et al. [42] to attach rendering geometries to our simulation models. In a pre-computation step each detail vertex is assigned to a surface triangle of the simulation model. To obtain barycentric coordinates, the vertex is projected onto its corresponding triangle using its interpolated normal. Finally, the distance to the surface point is stored. The deformed position of a detail vertex can be determined by interpolating the normal at the barycentric coordinates and moving the surface vertex at these coordinates along the normal by the stored distance.

## 6. Results

The simulations shown in this section were performed on two Intel X5650 processors with 2.66 GHz and six cores per processor. Our method was implemented using OpenMP to take advantage of the multi-core processor architecture. In all simulations we performed five iterations to reduce the strain energy and used a fixed time step size of  $\Delta t = 5$  ms. If not otherwise stated, the Saint Venant-Kirchhoff material is used. The volumetric models were visualized using the technique described in Section 5. Please see our accompanying video for the resulting animations.

*Performance.* Table 1 summarizes the number of vertices and elements as well as the timings of the simulations shown in Figure 1 and Figure 9. Note that the computation times do not include collision handling since this is not the focus of our paper. In the most complex simulation 100 Armadillos were falling through a funnel. To simulate elastic behavior of all Armadillos our approach required 152.41 ms per time step on average while the computation for a single Armadillo took 1.51 ms per step. The run-time scales linearly since a fixed number of iterations is used.

In our simulation we use a modified version of a Newton-Raphson solver to determine the position corrections for the energy reduction (see Section 3.2). To compare its convergence with the original Newton-Raphson solver we measured the total elastic energy over all tetrahedral elements in one simulation step of the dragon scene (see Figure 9). Figure 4 shows the results for different

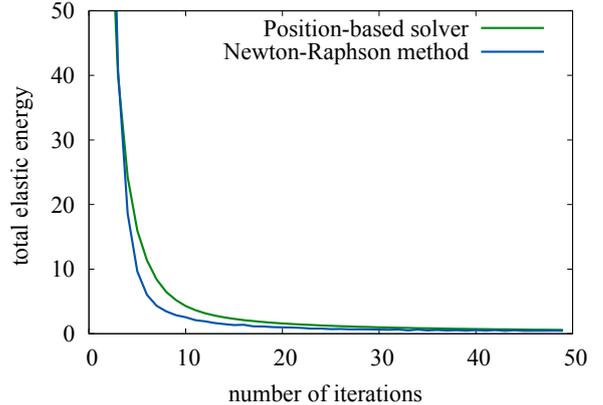


Figure 4: Convergence of the position-based energy reduction in one simulation step of the dragon scene (see Figure 9). This figure shows a comparison of the Newton-Raphson solver and its modified version used in our position-based approach.

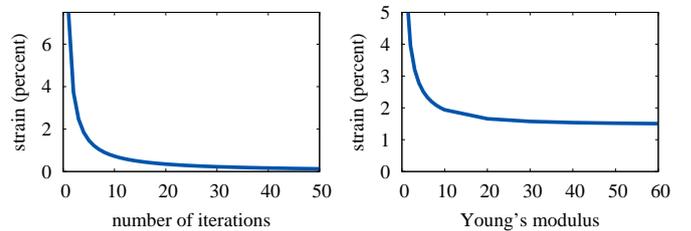


Figure 5: Total strain in percent of a beam stretched by a constant force of 100 N. Left: We varied the number of iterations while using a constant Young's modulus of  $k = 1000$ . Right: We used a fixed number of five iterations while varying the Young's modulus.

numbers of iterations. After the time integration performed in the first two lines of Algorithm 1 this energy is nearly 900. In the first five iterations both methods converge very fast. After that the energy decreases more slowly. In our simulations five iterations were enough to get realistic results. The figure shows that the Newton-Raphson solver converges slightly better than the modified version. However, in each iteration it has to solve a full linear system while the modified version only performs a single Gauss-Seidel iteration, where only the diagonal elements of the matrix are considered. Therefore, the modified version is significantly faster and the better convergence of the Newton-Raphson solver does not pay off. The convergence can also be improved considerably using a hierarchical solver as demonstrated by Müller [29]. This is especially advantageous when simulating almost stiff models.

In the next experiment we analyze the relationship between the total strain and the number of iterations as well as the stiffness parameter. For this experiment we stretched a simple beam model by fixing the top of the beam and applying a constant force of 100 N. We measured the total strain when the beam reaches its rest state while varying the number of iterations (see Figure 5, left) and the Young's modulus (see Figure 5, right). The results

| Model      | # vertices | # elements | elasticity | bending | total     |
|------------|------------|------------|------------|---------|-----------|
| Armadillos | 118000     | 371700     | 152.41 ms  | -       | 152.41 ms |
| Cloth      | 40401      | 80000      | 4.11 ms    | 2.62 ms | 6.73 ms   |
| Dragon     | 2603       | 7278       | 2.33 ms    | -       | 2.33 ms   |

Table 1: Complexity and timings of the simulations shown in Figure 1 and Figure 9. The time values represent the average computation times per simulation step required to reduce the elasticity and the bending energies.

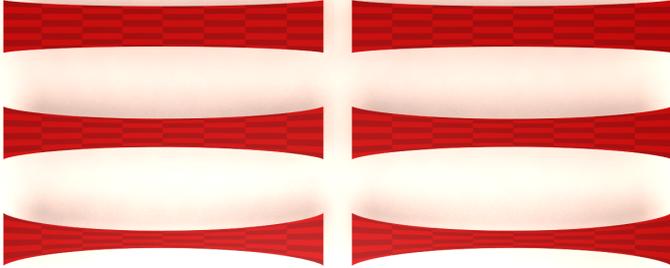


Figure 7: Lateral contraction of a stretched quadratic piece of cloth. Our approach (left) produces nearly the same results as a classical FEM simulation (right) for the Poisson ratios  $\nu = 0.1$ ,  $\nu = 0.3$  and  $\nu = 0.499$ .

show that the total strain converges fast in the first five iterations while the convergence slows down afterwards. In the second figure we can see the relationship between the Young’s modulus and the stiffness of the elastic material. Since the stiffness of the material generally depends on the number of iterations when using a position-based approach [15], our stiffness parameter  $k$  has not the same meaning as the Young’s modulus which is known from mechanics. However, it can be used to fine-tune the stiffness of the model.

*Stability.* In order to demonstrate the robustness of our approach, we ran two simulations with the Stanford Dragon. In the first one we set the y-coordinate of each vertex to zero while we inverted the whole model by a reflection in the second simulation. Our method was able to recover the model in both cases (see Figure 6) by using the inversion handling described in Section 4.2.

*Complex material behavior.* Due to the continuum-based formulation of our energy constraints, the energy gradients encode the characteristic deformation behavior of the simulated materials. Therefore, our approach can simulate physical phenomena like lateral contraction or anisotropy in contrast to previous position-based methods. This is demonstrated in the following.

We stretched a quadratic piece of cloth of size  $1\text{ m} \times 1\text{ m}$  with different Poisson ratios to compare the lateral contraction produced by our approach with a classical finite element simulation (see Figure 7). Since we employed a finite element method with an explicit time integration, we had to use a small time step size of  $\Delta t = 0.01\text{ ms}$  to keep the simulation stable. Our approach produces nearly the same results using a time step size of  $\Delta t = 5\text{ ms}$ . At the end of

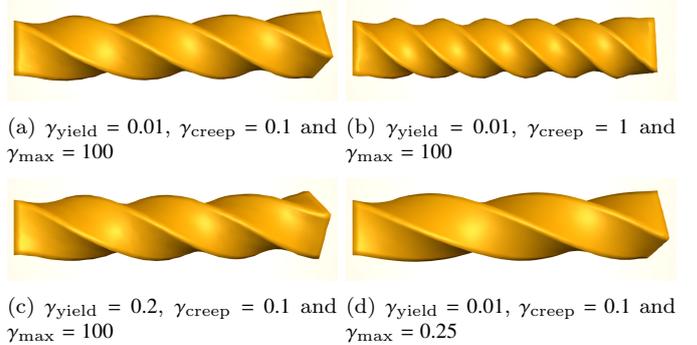


Figure 8: Plastic deformations when twisting a beam with different parameter sets.



Figure 9: Elastoplastic Stanford Dragon is deformed persistently due to the weight of a large sphere.

the simulation we measured the Hausdorff distance of the FEM solutions and ours. For all three Poisson ratios the maximum distance turned out to be smaller than  $4 \cdot 10^{-4}\text{ m}$  while the mean distance was even smaller than  $8 \cdot 10^{-6}\text{ m}$ . This demonstrates that the Poisson ratio is a meaningful mechanical parameter in terms of our approach.

Figure 8(a) shows the plastic deformation of a beam after being twisted by a direct manipulation of the vertices on the right. The usage of a larger creep parameter (see Figure 8(b)) resulted in a higher plastic strain and almost all elastic strain was absorbed when releasing the manipulated vertices. When increasing the parameter  $\gamma_{\text{yield}}$ , the plastic deformation begins later and less elastic strain is absorbed (see Figure 8(c)). In Figure 8(d) the maximal plastic strain was limited leading to a higher elastic strain



Figure 6: Dragon model recovers after being completely degenerate or inverted.

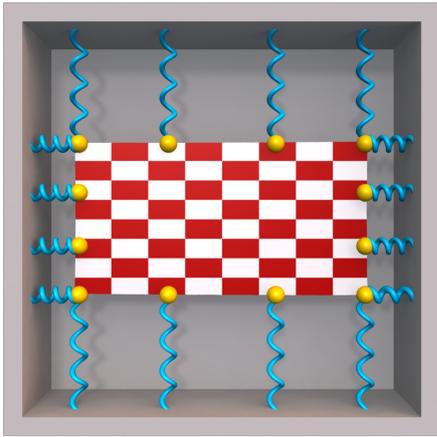


Figure 10: Horizontal and vertical spring forces of the same magnitude are applied to a quadratic piece of cloth. Using two different Young’s moduli  $k_{\text{weft}}$  and  $k_{\text{warp}}$  for the orthotropic cloth model results in an anisotropic stretching behavior.

at the end. A more complex example for plastic deformation is shown in Figure 9.

In order to demonstrate that our method can handle anisotropic material behavior, we stretched an orthotropic quadratic piece of cloth by horizontal and vertical spring forces of the same magnitude (see Figure 10). We used the orthotropic elasticity tensor defined in Section 4.3 with different Young’s moduli for the weft and warp direction. The figure shows the resulting anisotropic behavior of the cloth model.

Our approach supports different material models. We implemented the Saint Venant-Kirchhoff model and the Neo-Hookean model to demonstrate this. Figure 11 shows a comparison of both materials under compression. In the simulation a heavy ball is falling down on a deformable cube. This example demonstrates that the Neo-Hookean material can better resist a strong compression than the Saint Venant-Kirchhoff model. The reason for this is that the restorative force of the Saint Venant-Kirchhoff material reaches a maximum at a compression of approximately 58% and then it decreases (see Figure 2(a)). In contrast to that the restorative force of the Neo-Hookean model is not limited (see Figure 2(c)) and yields a significantly stronger reaction in case of an extreme compression.

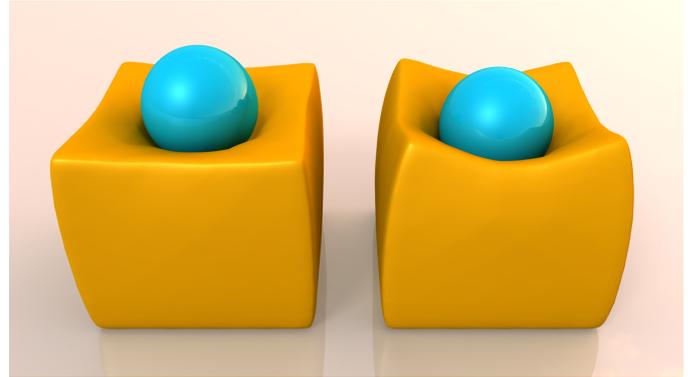


Figure 11: A soft cube is deformed by a heavy spherical object using different material models, but the same parameter set. In both images the point of maximal compression is captured. Left: Neo-Hookean material, right: Saint Venant-Kirchhoff material.

*Comparison with previous position-based methods.* We implemented position-based dynamics [21] with a parallelized Gauss-Seidel solver for a comparison with our method. The cloth model in Figure 1 was simulated using both approaches. The elasticity simulation needed 2.87 ms per step with position-based dynamics and 4.11 ms with our method. The position-based dynamics approach of Müller et al. is slightly faster. However, in contrast to our method this approach cannot handle anisotropic materials and has no parameter to control lateral contraction. In order to compare with shape matching, we implemented the parallel fast summation of Diziol et al. [18]. We used a simple beam model with 5000 vertices and 19845 tetrahedra to easily obtain a good path layout for their fast summation technique. Our elasticity simulation required 2.42 ms per step while shape matching needed 2.76 ms to achieve nearly the same stiffness. Shape matching performed a bit slower than our method. The reason for this is the costly polar decomposition that is required for shape matching. Another advantage of our method compared to shape matching is that shape matching cannot simulate effects like anisotropy and lateral contraction.

*Comparison with the classical finite element method.* In the following we want to compare the fundamental concepts of our position-based approach and the classical force-based finite element method to show the differences

between both techniques. In classical finite element simulations the elastic forces are determined by the negative gradient of the strain energy w.r.t. the vertex positions. When using an explicit time integration scheme, the resulting elastic forces are accumulated and then the velocities and positions are integrated. For the explicit integrator introduced in Section 3 and a single element our position-based approach delivers exactly the same velocities and positions when using a constant Lagrange multiplier of  $\lambda = -\Delta t^2$  and performing only one iteration of the position-based solver. The proof can be found in Appendix A. Note that for multiple elements we still get the same results with the constant Lagrange multiplier when computing the position corrections in a Jacobi fashion.

If the time step size is too large in a classical finite element simulation with an explicit integrator, the strain energy after the time step can be even larger than before and the system can become unstable. We call this the *overshooting problem*. Therefore, typically implicit time integration schemes are used in classical finite element simulations. In contrast to that our position-based approach uses an explicit time integration method. It determines a Lagrange multiplier per element which minimizes the element's energy approximately instead of using a fixed value of  $\lambda = -\Delta t^2$ . This means that our approach increases or decreases the finite element forces to avoid the overshooting problem and to minimize the strain energy locally. A global energy reduction is performed by computing the corresponding position corrections iteratively. Since we only modify the magnitude of the finite element forces but not their direction, the characteristic material behavior is preserved.

By adapting the finite element forces the position-based approach is able to perform a stable simulation of deformable solids with an explicit time integration scheme. This has the advantage that it allows a high performance. The disadvantage is that the stiffness of the model depends on the number of iterations and the time step size. Hence, only visual plausible results can be achieved. However, due to its stability and its performance the position-based approach is well-suited for the usage in computer games and other interactive applications.

## 7. Conclusion and Future Work

We presented a position-based approach based on a continuum mechanical formulation to simulate deformable solids and cloth. Our approach is fast, stable, easy to implement and controllable. In contrast to previous position-based methods it supports the simulation of complex physical phenomena like lateral contraction, anisotropy or elastoplasticity. However, our method also has some limitations. As for all position-based methods only visual plausibility can be achieved and the stiffness does not solely depend on the stiffness parameter (in our case the Young's modulus) but also on the number of iterations and the time step size. Another limitation of position-based approaches

is that they do not converge to a certain solution as the simulation mesh is refined. Therefore, adaptive time stepping and the usage of adaptive meshes are open problems for future work.

In future we plan to integrate the position-based fluids method of Macklin and Müller [22] in our framework in order to model materials with viscoelastic behavior. Moreover, a combination of both approaches would allow us to realize two-way coupling between deformable solids and fluids in a simple way. Finally, we plan to extend our approach by integrating a fracture criterion based on a stress analysis in order to model ductile fracture.

## Acknowledgments

The work of the authors was supported by the 'Excellence Initiative' of the German Federal and State Governments and the Graduate School CE at TU Darmstadt.

## References

- [1] Nealen A, Müller M, Keiser R, Boxerman E, Carlson M. Physically based deformable models in computer graphics. *Computer Graphics Forum* 2006;25(4):809–836.
- [2] Terzopoulos D, Witkin A. Deformable models. *IEEE CGA* 1988;8(6):41–51.
- [3] O'Brien JF, Hodgins JK. Graphical modeling and animation of brittle fracture. In: *Proc. SIGGRAPH*. ACM Press/Addison-Wesley Publishing Co.; 1999, p. 137–146.
- [4] O'Brien JF, Bargteil AW, Hodgins JK. Graphical modeling and animation of ductile fracture. *ACM Trans Graph* 2002;21(3):291–294.
- [5] James DL, Pai DK. Artdefo: Accurate real time deformable objects. In: *Proc. SIGGRAPH*. ACM Press/Addison-Wesley Publishing Co.; 1999, p. 65–72.
- [6] Etzmuß O, Keckeisen M, Straßer W. A fast finite element solution for cloth modelling. In: *Proc. Pacific Graphics*. IEEE Computer Society; 2003, p. 244–252.
- [7] Georgii J, Westermann R. Corotated finite elements made fast and stable. In: *Proc. Virtual Reality Interactions and Physical Simulations*. Eurographics Association; 2008, p. 11–19.
- [8] Dick C, Georgii J, Westermann R. A real-time multigrid finite hexahedra method for elasticity simulation using CUDA. *Simulation Modelling Practice and Theory* 2011;19(2):801–816.
- [9] Weber D, Bender J, Schnoes M, Stork A, Fellner D. Efficient GPU data structures and methods to solve sparse linear systems in dynamics applications. *Computer Graphics Forum* 2013;32(1):16–26.
- [10] Bender J, Deul C. Adaptive cloth simulation using corotational finite elements. *Computers & Graphics* 2013;37(7):820–829.
- [11] Hecht F, Lee YJ, Shewchuk JR, O'Brien JF. Updated sparse cholesky factors for corotational elastodynamics. *ACM Trans Graph* 2012;31(5):123:1–123:13.
- [12] Kaufmann P, Martin S, Botsch M, Gross M. Flexible simulation of deformable models using discontinuous galerkin fem. In: *Proc. Symposium on Computer Animation*. Eurographics Association; 2008, p. 105–115.
- [13] McAdams A, Zhu Y, Selle A, Empey M, Tamstorf R, Teran J, et al. Efficient elasticity for character skinning with contact and collisions. *ACM Trans Graph* 2011;30(4):37:1–37:12.
- [14] Chao I, Pinkall U, Sanan P, Schröder P. A simple geometric model for elastic deformations. *ACM Trans Graph* 2010;29(4).
- [15] Bender J, Müller M, Otaduy MA, Teschner M. Position-based methods for the simulation of solid objects in computer graphics. In: *EUROGRAPHICS 2013 State of the Art Reports*. Eurographics Association; 2013, p. 1–22.

[16] Müller M, Heidelberger B, Teschner M, Gross M. Meshless deformations based on shape matching. *ACM Trans Graph* 2005;24(3):471–478.

[17] Rivers AR, James DL. FastLSM: Fast lattice shape matching for robust real-time deformation. *ACM Trans Graph* 2007;26(3).

[18] Diziol R, Bender J, Bayer D. Robust real-time deformation of incompressible surface meshes. In: *Proc. Symposium on Computer Animation*. ACM; 2011, p. 237–246.

[19] Bender J, Weber D, Diziol R. Fast and stable cloth simulation based on multi-resolution shape matching. *Computers & Graphics* 2013;37(8):945–954.

[20] Müller M, Chentanez N. Solid simulation with oriented particles. *ACM Trans Graph* 2011;30(4):92:1–92:10.

[21] Müller M, Heidelberger B, Hennix M, Ratcliff J. Position based dynamics. *Journal of Visual Communication and Image Representation* 2007;18(2):109–118.

[22] Macklin M, Müller M. Position based fluids. *ACM Trans Graph* 2013;32(4):104:1–104:12.

[23] Deul C, Charrier P, Bender J. Position-based rigid body dynamics. In: *Proc. Computer Animation and Social Agents*. 2014, p. 1–9.

[24] Umetani N, Schmidt R, Stam J. Position-based elastic rod. In: *Proc. Symposium on Computer Animation*. Eurographics Association; 2014, p. 1–10.

[25] Stam J. Nucleus: Towards a unified dynamics solver for computer graphics. *IEEE International Conference on Computer-Aided Design and Computer Graphics* 2009;:1–11.

[26] Magnenat-Thalmann N, Volino P. From early draping to haute couture models: 20 years of research. *The Visual Computer* 2005;21:506–519.

[27] Baraff D, Witkin A. Large steps in cloth simulation. In: *Proc. SIGGRAPH*. ACM; 1998, p. 43–54.

[28] Thomaszewski B, Wacker M, Strasser W. A consistent bending model for cloth simulation with corotational subdivision finite elements. In: *Proc. Symposium on Computer Animation*. Eurographics Association; 2006, p. 107–116.

[29] Müller M. Hierarchical Position Based Dynamics. In: *Proc. Virtual Reality Interactions and Physical Simulations*. Eurographics Association; 2008, p. 1–10.

[30] Kelager M, Niebe S, Erleben K. A Triangle Bending Constraint Model for Position-Based Dynamics. In: *Proc. Virtual Reality Interactions and Physical Simulations*. Eurographics Association; 2010, p. 31–37.

[31] Thomaszewski B, Pabst S, Straßer W. Continuum-based strain limiting. *Computer Graphics Forum* 2009;28(2):569–576.

[32] Wang H, O’Brien J, Ramamoorthi R. Multi-resolution isotropic strain limiting. *ACM Trans Graph* 2010;29(6):156:1–156:10.

[33] Stern A, Desbrun M. Discrete geometric mechanics for variational time integrators. In: *ACM SIGGRAPH 2006 Courses*. ACM; 2006, p. 75–80.

[34] Bridson R, Fedkiw R, Anderson J. Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans Graph* 2002;21:594–603.

[35] Goldstein H, Poole C, Safko J. *Classical mechanics*. Addison Wesley; 2002.

[36] Bathe KJ. *Finite Element Procedures*. Prentice Hall; 1996.

[37] Sifakis E, Barbic J. FEM simulation of 3D deformable solids: A practitioner’s guide to theory, discretization and model reduction. In: *ACM SIGGRAPH 2012 Courses*. ACM; 2012, p. 20:1–20:50.

[38] Irving G, Teran J, Fedkiw R. Invertible finite elements for robust simulation of large deformation. In: *Proc. Symposium on Computer Animation*. Eurographics Association; 2004, p. 131–140.

[39] Gao Z, Kim T, James DL, Desai JP. Semi-automated soft-tissue acquisition and modeling for surgical simulation. In: *Proc. Conference on Automation Science and Engineering*. IEEE; 2009, p. 268–273.

[40] Müller M, Gross M. Interactive virtual materials. In: *Proc. Graphics Interface*. 2004, p. 239–246.

[41] Bergou M, Wardetzky M, Harmon D, Zorin D, Grinspun E.

A quadratic bending model for inextensible surfaces. In: *Proc. Symposium on Geometry processing*. Eurographics Association; 2006, p. 227–230.

[42] Kobbelt L, Vorsatz J, Seidel HP. Multiresolution hierarchies on unstructured triangle meshes. *Comput Geom Theory Appl* 1999;14(1-3):5–24.

## Appendix A. Comparison of our position-based approach with the classical FEM

In the following we analyze the relationship between the classical finite element method and our position-based approach considering a single element and using only one iteration step in the position-based solver.

When using the classical FEM, the elastic force for a particle  $i$  is determined by the negative gradient of the strain energy introduced in Section 4:

$$\mathbf{f}_i = -\frac{\partial E_s}{\partial \mathbf{x}_i}.$$

A time integration step with the explicit integration scheme introduced in Section 3 yields the following new velocities and positions for the particle:

$$\begin{aligned}\mathbf{v}_{i,\text{FEM}}^{n+1} &= \mathbf{v}_i^n + \Delta t w_i (\mathbf{f}_{\text{ext}}^n + \mathbf{f}_i^n) \\ \mathbf{x}_{i,\text{FEM}}^{n+1} &= \mathbf{x}_i^n + \Delta t \mathbf{v}_i^{n+1} = \mathbf{x}_i^n + \Delta t \mathbf{v}_i^n + \Delta t^2 w_i (\mathbf{f}_{\text{ext}}^n + \mathbf{f}_i^n).\end{aligned}$$

In contrast, the position-based approach first performs a time integration step in order to obtain the predicted position for particle  $i$ :

$$\begin{aligned}\mathbf{v}_i^{n+1} &= \mathbf{v}_i^n + \Delta t w_i \mathbf{f}_{\text{ext}}^n \\ \mathbf{x}_i^{n+1} &= \mathbf{x}_i^n + \Delta t \mathbf{v}_i^{n+1} = \mathbf{x}_i^n + \Delta t \mathbf{v}_i^n + \Delta t^2 w_i \mathbf{f}_{\text{ext}}^n.\end{aligned}$$

Then a position correction  $\Delta \mathbf{x}_i = w_i \lambda \nabla_{\mathbf{x}_i} C(\mathbf{x})$  is determined which gives us the final positions

$$\mathbf{x}_{i,\text{PBD}}^{n+1} := \mathbf{x}_i^n + \Delta t \mathbf{v}_i^n + \Delta t^2 w_i \mathbf{f}_{\text{ext}}^n - w_i \lambda \mathbf{f}_i^n$$

since in our case  $\nabla_{\mathbf{x}_i} C(\mathbf{x}) = \frac{\partial E}{\partial \mathbf{x}_i} = -\mathbf{f}_i$ . Finally, the velocities are updated (see line (10) in Algorithm 1) which yields

$$\mathbf{v}_{i,\text{PBD}}^{n+1} := \frac{1}{\Delta t} (\mathbf{x}_{i,\text{PBD}}^{n+1} - \mathbf{x}_i^n) = \mathbf{v}_i^n + \Delta t w_i \mathbf{f}_{\text{ext}}^n - \frac{1}{\Delta t} w_i \lambda \mathbf{f}_i^n.$$

This shows that for  $\lambda = -\Delta t^2$  our position-based approach is equivalent to the classical FEM when considering one element and performing one iteration of the position-based solver.