ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2022
D. L. Michels and S. Pirk
(Guest Editors)

Volume 41 (2022), Number 8

Physically Based Shape Matching

Matthias Müller Miles Macklin Nuttapong Chentanez Stefan Jeschke

NVIDIA

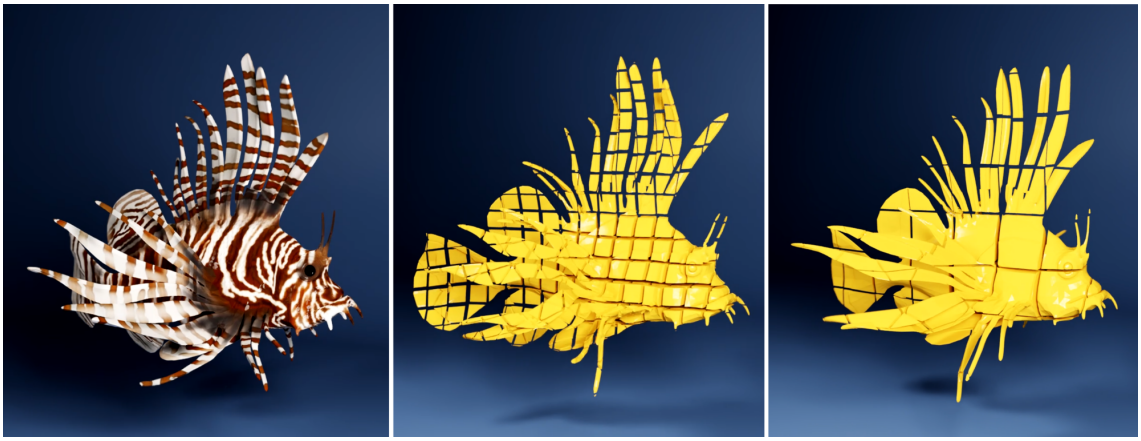

Figure 1: *Our method allows the direct simulation of a cut-cell mesh comprised of arbitrarily shaped elements. The simulation mesh matches the input surface mesh perfectly. Even at low resolution, the lion fish deforms in a plausible way.*

Abstract

The shape matching method is a popular approach to simulate deformable objects in interactive applications due to its stability and simplicity. An important feature is that there is no need for a mesh since the method works on arbitrary local groups within a set of particles. A major drawback of shape matching is the fact that it is geometrically motivated and not derived from physical principles which makes calibration difficult. The fact that the method does not conserve volume can yield visual artifacts, e.g. when a tire is compressed but does not bulge.

In this paper we present a new meshless simulation method that is related to shape matching but derived from continuous constitutive models. Volume conservation and stiffness can be specified with physical parameters. Further, if the elements of a tetrahedral mesh are used as groups, our method perfectly reproduces FEM based simulations.

Keywords: finite element method, physically-based animation, elasticity, real-time physics

CCS Concepts

• **Computing methodologies** → **Physical simulation;**

1. Introduction

Simulating deformable objects has a long history in computer graphics. Applications are flesh simulations on characters or rubbery objects like tires. A large variety of methods have been proposed in this field. Many of those methods are mesh-based meaning the simulation is performed on a volumetric mesh enclosing an object. Since creating such a mesh is a difficult problem, mesh generation is a large field in itself.

Meshless methods have been introduced to avoid this problem. Here, objects are typically sampled by particles. The particles are held together by constraints or forces acting on local particle groups. Other than not requiring a mesh, they have a number of additional advantages. For instance, topological changes can be handled by simply including or excluding particles from groups. When large plastic deformations occur, objects can be re-sampled easily and the groups re-created based on particle distances.

Mass spring networks are a simple example. Here, the interac-

tions of the particles are modeled with springs. A difficulty of this method is its dependency on the tessellation of the simulation mesh. Also, the stiffnesses of the springs have to be tuned to get desired overall behavior. The main problem with the method is the fact that objects do not recover from inverted states.

Müller et al. [MHTG05] introduced a meshless simulation method based on shape matching to solve this problem. Besides resolving inversions, as a position based method it is unconditionally stable. A major drawback of the approach is that it is based on a geometric constraint and not derived from the discretization of a continuous model. The method does not conserve volume which means, when an object is stretched, it does not contract in the orthogonal directions. It is also difficult to specify an elastic modulus that yields a desired behavior.

The goal of this project was to find a method that is as simple and robust as shape matching but derived from physical principles. Shape matching is a positional constraint. We also use positional constraints but derive them from energy terms of existing models as proposed by Macklin et al. [MM21]. As Macklin et al. we use the extended position based dynamics method (XPBD) to handle constraints which yields an unconditionally stable simulation that can be implemented with a few lines of code.

While Macklin et al. demonstrate their method using a Neo-Hookean model, we show that the approach works for other constitutive models as well. To do this, we formulate the Hookean model as a single positional constraint.

Using an arbitrary number of particles per element introduces so-called zero-energy modes, motions of particles that go unnoticed by the model and yield bad visual artifacts. In [KBFF*21], Kugelstadt et al. suppress zero energy modes by introducing an additional energy term. In addition to being non-physical, their stiff penalty term introduces numerical challenges. In contrast, we use a projection step similar to shape matching to filter out these modes. In summary, our contributions are

- a physically based simulation method that can handle arbitrary groups of particles as elements,
- formulating the Hookean model with a single position based constraint and
- a simple and stable way to filter zero-energy modes.

2. Related Work

The simulation of deformable solids has been a subject of extensive research in computer graphics for the last few decades. An overview of research in this area up to 2005 and an explanation of the underlying principles can be found in [NMK*06]. Kim and Eberle provide a more recent survey of the field in their Siggraph course notes [KE20].

The Finite Element Method (FEM) is one of the most popular methods used for the simulation of soft bodies in computer graphics [BWHT07, KMBG08, WJST15]. As we already mentioned, a major challenge here is to generate a consistent volumetric mesh, typically tetrahedral or hexahedral.

Point based methods or meshless methods solve this problem. Currently, the most popular method in computer graphics is the Material Point Method (MPM) [Jia, HFG*18, WCL*20, WDG*19, DHW*19]. It is a hybrid method in that it samples the quantities, stored on particles to a regular background grid, solves the equations on the grid and samples the resulting field on the particles.

Shape matching [MHTG05] is a position based method. An overview of position based dynamics (PBD) can be found in [BMM17]. Extended position based dynamics (XPBD) [MMC16] is a key improvement over the original method as it introduces the concept of physical stiffness which is independent of time step size and iteration count. Macklin and Müller [MM21] showed that continuous models such as the Neo-Hookean model [SGK18] can be formulated as constraints in the XPBD framework which we use to make shape matching physically based.

Shape matching has been extended in various ways before e.g. to simulate ductile fracture [JML*16] to handle large plastic deformation using cluster resampling [FJL*17], [CMM16]. These extensions are orthogonal to our work and will benefit from a physically based variant.

3. Method

3.1. Constrained Based Formulation

We use the extended position based dynamics framework (XPBD) for simulation. Here, energies are replaced by compliant constraints. Each constraint is associated with a scalar compliance parameter α which is the inverse of physical time-step-independent stiffness. Infinitely stiff materials can be handled stably by setting the compliance to zero. While the actual stiffness of objects still depends on solver convergence, the method does not require stiffness tuning and does not have instabilities near or at the inextensible or incompressible state.

Macklin and Müller [MM21], showed how a Neo-Hookean model can be reproduced one to one in the XPBD framework using two positional constraints. In the following section we state the derivation of their method to make this paper self-contained.

3.2. Neo-Hookean Model

For the Neo-Hookean model with the potential energy density

$$\Psi_{\text{Neo}} = \frac{\lambda}{2}(\det(\mathbf{F})-\gamma)^2 + \frac{\mu}{2}\left(\text{tr}(\mathbf{F}^T\mathbf{F})-3\right) \quad (1)$$

$$= \Psi_H + \Psi_D, \quad (2)$$

the hydrostatic energy density Ψ_H and the deviatoric energy density Ψ_D can be replaced by the two constraints for each element in a mesh

$$C_H(\mathbf{F}) = \det(\mathbf{F})-\gamma \text{ and} \quad (3)$$

$$C_D(\mathbf{F}) = \sqrt{\text{tr}(\mathbf{F}^T\mathbf{F})}, \quad (4)$$

with corresponding compliance parameters

$$\alpha_H = \frac{1}{\lambda V_e} \text{ and} \quad (5)$$

$$\alpha_D = \frac{1}{\mu V_e}. \quad (6)$$

Here, $\mathbf{F}$ is the deformation gradient, λ and μ the Lamé parameters and V_e the volume of a finite element of a mesh. In the original

Neo-Hookean model γ is 1. To make the model rest stable, Smith et al. [SGK18] proposed to set $\gamma = 1 + \frac{\mu}{\lambda}$. Constraints are defined via scalar constraint functions C which are zero iff the constraint is satisfied.

The equivalency can be proved by considering the relation of energies and constraints in the XPBD approach. The XPBD constraint projection is derived from a constraint based energy potential

$$U(\mathbf{x}) = \frac{1}{2}\alpha^{-1}C(\mathbf{x})^2 \tag{7}$$

where the compliance α is the inverse of stiffness. If we substitute the general constraint function C and compliance parameter α in Equation (7) with the specific Neo-Hookean quantities defined in Equations (3) - (6), we recover the original Neo-Hookean energy potentials without the constant term -3 on the very right. However, subtracting a constant from an energy does not change the resulting forces.

The energy Ψ_{Neo} in Equation (2) is an energy *density* while $U(x)$ in Equation (7) is a regular energy. If we consider $\mathbf{F}$ to be constant within an element then integrating the energy density over the element amounts to a multiplication with the element's volume V_e. This is the reason why V_e appears in Equations (5) and (6).

3.3. Hookean Model

In addition to the Neo-Hookean model, we will derive the constraint functions for a Hookean model as well. Bender et al. [BKCW14] derived a constraint function for the Hookean model within the original PBD framework. While their constraint drives elements towards their rest shapes, the constraint based model is not equivalent to the original model. First we follow Bender et al. and compute strains from the deformation gradient using the Green-Lagrange strain tensor

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left(\mathbf{F}^T\mathbf{F} - \mathbf{I}\right), \tag{8}$$

where $\mathbf{I}$ denotes the identity matrix. Next we apply Hooke's law to compute stresses from strains as

$$\mathbf{S} = \mathbf{C}\boldsymbol{\varepsilon}, \tag{9}$$

where $\mathbf{C}$ is the fourth order elasticity tensor. The scalar elastic energy density due to the deformation is given by

$$W = \frac{1}{2}\boldsymbol{\varepsilon} : \mathbf{S} = \frac{1}{2}\boldsymbol{\varepsilon} : \mathbf{C}\boldsymbol{\varepsilon}, \tag{10}$$

with the inner product $\mathbf{A} : \mathbf{B} = \sum_{i,j} a_{i,j}b_{i,j}$.

Bender et al. now define the constraint function to be $C_{\text{Hooke}}(\mathbf{F}) = W(\mathbf{F})$ which is satisfied when the deformation energy is zero. To introduce stiffness, they use the non-physical method of PBD which simply scales the correction vectors.

We will now describe how we solve this problem.To get matching forces, we need a relation of the elastic modulus E and the compliance parameter α used in XPBD. An important observation is that E can be factored out of the energy so we can write $W = E\hat{W}$ with $\hat{W} = W|_{\text{E=1}}$ (see Appendix 6.1). If we now define the constraint function

$$C_{\text{Hooke}}(\mathbf{F}) = \sqrt{2\hat{W}(\mathbf{F})}. \tag{11}$$

and use the compliance

$$\alpha = \frac{1}{EV_e}, \tag{12}$$

we match the original Hookean model. This can be seen by substituting the constraint function and compliance parameter in Equation (7). We parametrized the Hookean model using Young's modulus and the Poisson ratio. However, other pairs of parameters can be used such as the the Lamé parameters in the Neo-Hookean case. We chose the former possibility because we believe that it is simpler and more intuitive to tune.

3.4. Meshless Discretization

We represent an object as a set of particles with positions $\mathbf{x}_i$ and a set of local particle groups which are the equivalents of the elements in a mesh. The core of a discretization is the estimation of the deformation gradient $\mathbf{F}$ based on the particle positions $\mathbf{x}_i$ in a meshless approach or the nodal positions in a mesh based approach.

For a tetrahedron with four nodal positions, the deformation gradient is uniquely defined. However, for an arbitrary number of particles in a group, the definition is not unique anymore. With more than four points in a group, the problem of finding $\mathbf{F}$ is overconstrained. For this reason, a least squares fit is typically applied. We select the one used in shape matching [MHTG05] where $\mathbf{F}$ is defined as

$$\mathbf{F} = \left(\sum_{i=1}^{n} m_i\mathbf{r}_i\bar{\mathbf{r}}_i^T\right)\left(\sum_{i=1}^{n} m_i\bar{\mathbf{r}}_i\bar{\mathbf{r}}_i^T\right)^{-1} = \mathbf{P}\mathbf{Q}^{-1}, \tag{13}$$

where n is the number of particles in the set, $\mathbf{r}_i = \mathbf{x}_i - \mathbf{x}_{cm}$, $\bar{\mathbf{r}}_i = \bar{\mathbf{x}}_i - \bar{\mathbf{x}}_{cm}$. The center of mass is computed as

$$\mathbf{x}_{cm} = \frac{\sum_{i=1}^{n} m_i\mathbf{x}_i}{\sum_{i=1}^{n} m_i} \tag{14}$$

and analogous for $\bar{\mathbf{x}}_{cm}$. For a fixed connectivity, the rest quantities $\mathbf{Q}^{-1}$ and $\bar{\mathbf{x}}_{cm}$ are constant and can be pre-computed.

If we plug our definition of $\mathbf{F}$ into the constraint functions defined in the previous sections, we get constraints on the positions of the particles within a group.

To derive the compliance α via Equations (5) and (6), we need an estimation for the volume represented by the group. For a regular sampling, dividing the volume of the object by the number of groups is a good estimate. In a more general setting one could use the volume of a sphere of radius $\bar{r} = \sum_{i=1}^{n} |\bar{\mathbf{r}}_i|/n$. If there is one group per particle, another possibility is to compute the density ρ_i at each particle using a normalized kernel as in [MKN*04] and define the volume V_i of the group associated with particle i to be m_i/ρ_i.

We used yet another approach for the simulation shown in Fig. 1. We cut the volume defined by the input mesh geometrically using a regular grid as stencil. The resulting cut-cell mesh has regular hex-cells in the interior and arbitrarily shaped cells on the boundary. The boundary cells are defined by the clipped part of the original surface that lies inside the corresponding cell and the clipped parts of the faces of the regular mesh lying inside the cell. This way the boundary cells perfectly match the volume defined by the input

surface mesh. We then turn all the vertices of the original input surface as well as all the vertices created by the clipping process into particles.

The clipping can yield multiple disconnected volumes per boundary cell. We create one group for each disconnected piece and one group for each interior cell using all vertices of the clipped mesh lying inside the cell as adjacent particles. In this case, the volume of the groups can be computed precisely using the surface of each cell.

3.5. Filtering Zero-Energy Modes

Simulating objects solely with the constraints described above yields disturbing artifacts. Particles can move freely in an unconstrained way causing an object to deform in arbitrary ways to random chaotic shapes.

The reason is that if there are more than four particles in a group, the group has more degrees of freedom than the deformation gradient given by Equation (13). In other words, there are sub-spaces of particle configurations that yield the same deformation gradient. Therefore, our constraints cannot distinguish between configurations of the same subspace and the particles can move freely within them. These motions which are invisible to the energy computation are called zero-energy modes.

This is an important problem that also appears in hybrid particle methods that use a background grid because the particles represent more degrees of freedom than the grid cells. FLIP [ZB05] which is the predecessor of MPM is a hybrid method that introduces zero-energy modes. The reason is that there are many arrangements of particle velocities that, when sampled on the grid, are divergence free on the grid. Since FLIP only computes velocity *corrections*, the zero-energy motions remain. This problem is usually reduced by using a hybrid FLIP - PIC method because PIC removes all zero-energy modes. The same is true for traditional MPM. It happens when quantities of the particles are *overwritten* by smoothed values interpolated from the grid cells. Going to the grid and back smoothes out all the additional degrees of freedom of the particles.

We carry this idea over to the grid-less method as follows: as an additional step of constraint projection, we re-compute $\mathbf{F}$ and $\mathbf{x}_{cm}$ using Equations (13) and (14) respectively which now reflect the application of the material model. Then we replace the positions of all particles in a group as

$$\mathbf{x}_i \leftarrow \mathbf{x}_{cm} + \mathbf{F}\,\bar{\mathbf{r}}_i. \tag{15}$$

In the case of a tetrahedron or four particles, this update does not change the particle positions. In the over-constrained configuration however, it replaces the particle positions with the configuration matching $\mathbf{F}$ that is closest to the rest state modulo rotation and translation. This step replaces the shape-matching projection step but uses a physically derived deformation gradient instead of a rotation matrix.

3.6. Algorithm

Algorithm 1 shows the single iteration XPBD simulation loop. We run s substeps in each simulation step. Within each substep we perform the particle prediction, a single solver iteration and the velocity update of XPBD. To simulate damping, we iterate through all groups again and modify the updated velocities as described in section 3.7.

Algorithm 1: XPBD solver

```
while simulating do
    h ← Δt/numSubsteps;
    for s substeps do
        for n particles do
            x_prev ← x;
            v ← v + h f_ext/m;
            x ← x + h v;
        end
        for all groups g do
            project(g, x_1, ..., x_{n_g});
        end
        for n particles do
            v ← (x − x_prev)/h;
        end
        for all groups g do
            applyDamping(g, v_1, ..., v_{n_g});
        end
    end
end
```

It might seem that a single solver iteration would yield an inaccurate simulation. However, in [MSL*19] the authors showed that for a fixed computation budget per time step $b = s \cdot n$, where s is the number of substeps and n the number of solver iterations per substep, $s = b$ and $n = 1$ yields the most accurate result and converges fastest. It also resolves the most temporal detail and introduces the smallest amount of damping. A large number of iterations might solve the implicit equations more accurately. However, even the analytic solution of an implicit integration scheme is just an approximation of the analytic trajectory which becomes worse with increasing time step size. In other words it is better to solve more accurate equations approximately than one inaccurate equation close to convergence.

The projection function project$(g, \mathbf{x}_1, \ldots, \mathbf{x}_{n_g})$ referred to in Algorithm 1 applies positional corrections to the particles of a group g in order to solve the group constraint. Based on XPBD, a Lagrange multiplier λ per group is computed as

$$\lambda = \frac{-C(\mathbf{x})}{\sum_{i=1}^{n} w_i |\nabla_{\mathbf{x}_i} C(\mathbf{x})|^2 + \alpha/h^2}, \tag{16}$$

where h is the substep size, α the compliance (inverse stiffness) we discussed in the previous section, the w_i the inverse masses of the particles and $\mathbf{x}$ the concatenation of the positions of the particles in the group. The numerator, i.e. the negative value of the constraint function, can be computed by first computing the deformation gradient $\mathbf{F}$ using Equation (13) and then evaluating the constraint functions defined in terms of $\mathbf{F}$. In the denominator, the gradients of the constraint functions with respect to the particle positions are needed

as well. Fortunately, they turn out to be quite simple expressions. With $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3]$ they evaluate to

$$\nabla_{\mathbf{x}_i} C_H(\mathbf{x}) = m_i [\mathbf{f}_2 \times \mathbf{f}_3, \mathbf{f}_3 \times \mathbf{f}_1, \mathbf{f}_1 \times \mathbf{f}_2] \mathbf{Q}^{-T} \bar{\mathbf{r}}_i, \quad (17)$$

$$\nabla_{\mathbf{x}_i} C_D(\mathbf{x}) = \frac{m_i}{r} [\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3] \mathbf{Q}^{-T} \bar{\mathbf{r}}_i, \text{ and} \quad (18)$$

$$\nabla_{\mathbf{x}_i} C_{\text{Hooke}}(\mathbf{x}) = \frac{m_i}{C_{\text{Hooke}}(\mathbf{x})} \mathbf{S}\mathbf{F}\mathbf{Q}^{-T} \bar{\mathbf{r}}_i, \quad (19)$$

where $r = \sqrt{|\mathbf{f}_1|^2 + |\mathbf{f}_2|^2 + |\mathbf{f}_3|^2}$. The matrix $\mathbf{Q}$ is defined in Equations (13). Once the per-group multiplier λ is known, we can compute the individual updates for the particles contained in the group as

$$\mathbf{x}_i \leftarrow \mathbf{x}_i + \lambda w_i \nabla_{\mathbf{x}_i} C(\mathbf{x}). \quad (20)$$

The sequential version of XPBD applies projections immediately to each group before proceeding to the next corresponding to the nonlinear Gauss-Seidel method. It is more stable then methods which ignore updates of adjacent groups because it prevents overshooting but it makes the simulation dependent on constraint ordering. Randomizing the order but keeping it constant throughout the simulation prevents potential artifacts.

3.7. Damping

Damping is applied during a velocity pass after the position solve in Algorithm 1. We use the method proposed by Müller et al. [MHR06] which is shown in Algorithm 2.

Algorithm 2: ApplyDamping

$\mathbf{x}_{\text{cm}} \leftarrow (\sum_i \mathbf{x}_i m_i) / (\sum_i m_i)$;
$\mathbf{v}_{\text{cm}} \leftarrow (\sum_i \mathbf{v}_i m_i) / (\sum_i m_i)$;
$\mathbf{L} \leftarrow \sum_i (\mathbf{r}_i \times (m_i \mathbf{v}_i))$;
$\mathbf{I} \leftarrow \sum_i \left(\mathbf{R}_i \mathbf{R}_i^T m_i \right)$;
$\omega \leftarrow \mathbf{I}^{-1} \mathbf{L}$;
for *all particles i* **do**
  $\bar{\mathbf{v}}_i \leftarrow \mathbf{v}_{\text{cm}} + \omega \times \mathbf{r}_i$
  $\mathbf{v}_i \leftarrow \mathbf{v}_i + \min(c\Delta t, 1)(\bar{\mathbf{v}}_i - \mathbf{v}_i)$
end

Here $\mathbf{r}_i = \mathbf{x}_i - \mathbf{x}_{\text{cm}}$, $\mathbf{R}_i$ the matrix for which $\mathbf{R}_i \mathbf{v} = \mathbf{r}_i \times \mathbf{v}$ and c the damping coefficient. We use the fact that a fully damped group of connected particles moves like a rigid body with velocity $(\mathbf{v}_{\text{cm}}, \omega)$. The individual velocities of the particles are driven toward the global motion in a time step independent way using the damping coefficient c. XPBD allows clamping the correction to not overshoot making the simulation stable for arbitrary values of c.

4. Results

We ran all our tests on a single thread of a Core-i7-9700K CPU at 3.6 GHz. Parallelizing our method is straight forward using a Jacobi-style solver as described in [BMM17].

As a first experiment we stretched a bar. The bar is composed of hexahedral elements with randomized shapes. The result of the experiment is shown in Figure 2. The blue bar on the left shows

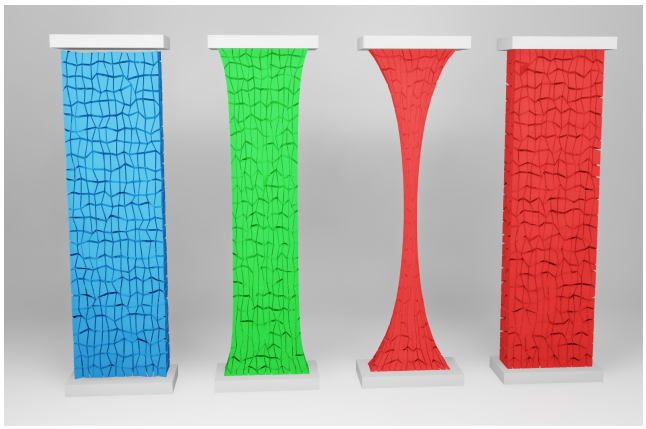

Figure 2: *Stretching experiment. From left to right: Shape Matching, Neo-Hookean, Hookean ($\nu = 0.49$) and Hookean ($\nu = 0.0$).*

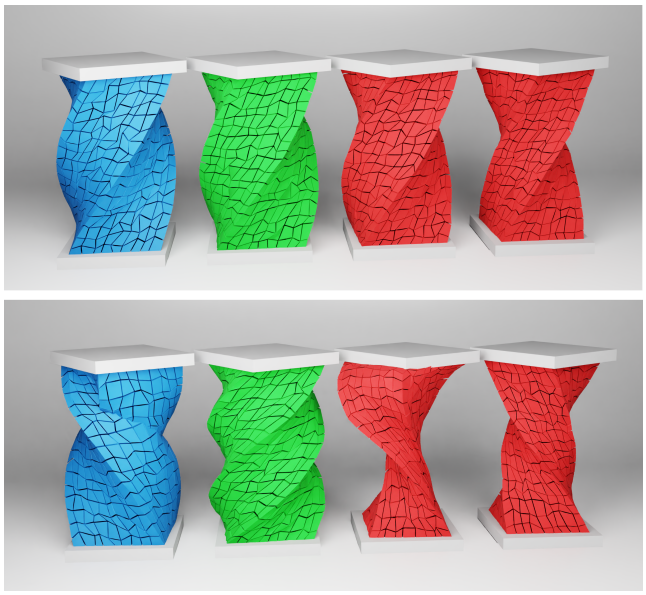

Figure 3: *Twisting experiment: The Neo-Hookean model is the most stable among the four. It keeps the correct shape while the other three blocks collapse.*

that shape matching does not conserve volume because it does not generate orthogonal forces. To simulate the green bar we used the Neo-Hookean model. The bar shows the characteristic shape of a volume conserving hyperelastic material. The volume is conserved within 3 percent. The red bars are simulated using the Hookean model with a Poisson ratio of 0.49 and 0.0. The left bar shows that the Hookean model overcompensates volume conservation. This artifact is a result of the way volume conservation is modelled as off-diagonal entries in the constitutive matrix which is an approximation valid for small deformations only.

Choosing a Poisson ratio of 0.0 matches the behavior of shape matching. The simulation of all bars takes 170 milliseconds per frame. We used 1700 elements and 2300 particles per bar.

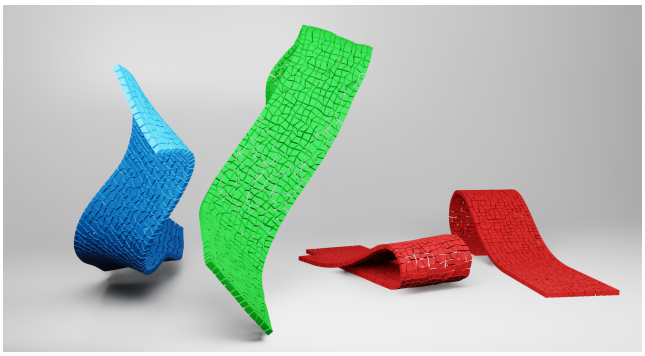

Figure 4: *The simulation of thin layers with user interaction demonstrates the stability of the position based formulation.*

In the second experiment shown in Figure 3 we twisted the bar by 270 degrees. Shape matching as well as the Neo-Hookean model handle this scenario well. The Neo-Hookean model produces a slightly smoother shape. The Hookean model fails because it does not generate forces that restore inverted elements. Here we used 1400 elements and 1800 particles per bar which resulted in 140 milliseconds per frame.

Figure 4 shows an experiment in which we created thin shells. No matter how fast the user drags them, the simulation remains stable due to the fact that we use a position based compliant formulation. The sheets are composed of 1060 elements and 2300 particles. This simulation runs at 100 milliseconds per frame.

For the simulation shown in Figure 1 we used a cut-cell mesh as described in Section 3.4. There is one particle group for each disconnected cell. We compute the masses of the particles by distributing the masses of the cells evenly among the adjacent particles. The mass of a cell can be computed by multiplying the element's volume with the density of the object. For this demo, we cut the surface mesh with 20k triangles into 840 cells. The resulting simulation mesh perfectly matches the input surface. For the simulation we applied the Neo-Hookean material model. This simulation takes 82 milliseconds per frame. The mesh shown on the right of Figure 1 shows a lower resolution mesh of only 195 cells. Even at this low resolution, the simulation still creates a plausible behavior. The speed of the simulation is comparable to the higher resolution case because each individual cell has now a larger number of adjacent particles.

5. Conclusion and Future Work

We have presented a mesh-less method to simulate elastic solids. It resembles shape matching but is derived from physical principles. We demonstrated the approach using a Hookean and a Neo-Hookean constitutive model.

The linear filter to remove the zero-energy modes introduces so-called shear locking . The resulting artifact only shows for rather soft objects which exhibit large deformations. A way to fix this problem would be to use non-linear shape matching as discussed in the original shape matching paper [MHTG05]. The cut-cell mesh we use perfectly matches the input mesh. However, often, an approximate mesh is sufficient for simulation. Therefore we are looking into ways to simplify the cells while keeping the mesh consistent. Another line of research will be to investigate the application of our method to objects under large plastic deformation or tearing and cutting.

6. Appendix

6.1. Hookean Constitutive Matrix

For isotropic materials and if the stress and strain tensors are written as one dimensional vectors

$$\varepsilon = [\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{zz}, \varepsilon_{xy}, \varepsilon_{yz}, \varepsilon_{zx}]^T \text{ and} \tag{21}$$

$$\sigma = [\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{yz}, \sigma_{zx}]^T, \tag{22}$$

the fourth order constitutive matrix of a Hookean model can be written as a regular matrix

$$\mathbf{C} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-2\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & 1-2\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & 1-2\nu \end{bmatrix}, \tag{23}$$

where E is Young's modulus, ν the Poisson ratio and $\sigma = \mathbf{C}\varepsilon$. As Equation (23) shows, Young's modulus E can be factored out and interpreted as a stiffness parameter such that $\sigma = E\,\hat{\mathbf{C}}\varepsilon$, where $\hat{\mathbf{C}} = \mathbf{C}|_{E=1}$.

6.2. Implementation Notes

The entries of the two matrices in $\mathbf{P}$ and $\mathbf{Q}$ Equation (13) might get very small for dense samplings. For the inversion of $\mathbf{Q}$, the determinant has to be computed which is even smaller and might fall out of the range of single precision floating point numbers. To fix this problem, we compute a normalization number $s = 1/\left(\sum_{i,j} q_{ij}\right)$, where the q_{ij} are the entries of $\mathbf{Q}$. We then multiply $\mathbf{Q}$ by this number before inversion. In this case, $\mathbf{P}$ as well as all the gradients have to be multiplied by s as well.

References

[BKCW14] BENDER J., KOSCHIER D., CHARRIER P., WEBER D.: Position-based simulation of continuous materials. *Computers & Graphics 44*, 0 (2014), 1 – 10. doi:http://dx.doi.org/10.1016/j.cag.2014.07.004. 3

[BMM17] BENDER J., MÜLLER M., MACKLIN M.: A Survey on Position Based Dynamics. In *EG 2017 - Tutorials* (2017), Bousseau A., Gutierrez D., (Eds.), The Eurographics Association. doi:10.2312/egt.20171034. 2, 5

[BWHT07] BARGTEIL A. W., WOJTAN C., HODGINS J. K., TURK G.: A finite element method for animating large viscoplastic flow. In *ACM SIGGRAPH 2007 Papers* (New York, NY, USA, 2007), SIGGRAPH '07, Association for Computing Machinery, p. 16–es. URL: https://doi.org/10.1145/1275808.1276397, doi:10.1145/1275808.1276397. 2

[CMM16] CHENTANEZ N., MÜLLER M., MACKLIN M.: Real-time simulation of large elasto-plastic deformation with shape matching. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Goslar, DEU, 2016), SCA '16, Eurographics Association, p. 159–167. 2

[DHW*19] DING M., HAN X., WANG S., GAST T. F., TERAN J. M.: A thermomechanical material point method for baking and cooking. *ACM Trans. Graph. 38*, 6 (nov 2019). URL: https://doi.org/10.1145/3355089.3356537, doi:10.1145/3355089.3356537. 2

[FJL*17] FALKENSTEIN M., JONES B., LEVINE J. A., SHINAR T., BARGTEIL A. W.: Reclustering for large plasticity in clustered shape matching. In *Proceedings of the Tenth International Conference on Motion in Games* (New York, NY, USA, 2017), MIG '17, Association for Computing Machinery. URL: https://doi.org/10.1145/3136457.3136473, doi:10.1145/3136457.3136473. 2

[HFG*18] HU Y., FANG Y., GE Z., QU Z., ZHU Y., PRADHANA A., JIANG C.: A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Trans. Graph. 37*, 4 (jul 2018). URL: https://doi.org/10.1145/3197517.3201293, doi:10.1145/3197517.3201293. 2

[Jia] 2

[JML*16] JONES B., MARTIN A., LEVINE J. A., SHINAR T., BARGTEIL A. W.: Ductile fracture for clustered shape matching. In *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2016), I3D '16, Association for Computing Machinery, p. 65–70. URL: https://doi.org/10.1145/2856400.2856415, doi:10.1145/2856400.2856415. 2

[KBFF*21] KUGELSTADT T., BENDER J., FERNÁNDEZ-FERNÁNDEZ J. A., JESKE S. R., LÖSCHNER F., LONGVA A.: Fast corotated elastic sph solids with implicit zero-energy mode control. *Proc. ACM Comput. Graph. Interact. Tech. 4*, 3 (Sept. 2021). URL: https://doi.org/10.1145/3480142, doi:10.1145/3480142. 2

[KE20] KIM T., EBERLE D.: Dynamic deformables: Implementation and production practicalities. In *ACM SIGGRAPH 2020 Courses* (New York, NY, USA, 2020), SIGGRAPH 2020, Association for Computing Machinery. URL: https://doi.org/10.1145/3388769.3407490, doi:10.1145/3388769.3407490. 2

[KMBG08] KAUFMANN P., MARTIN S., BOTSCH M., GROSS M.: Flexible simulation of deformable models using discontinuous galerkin fem. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Goslar, DEU, 2008), SCA '08, Eurographics Association, p. 105–115. 2

[MHR06] MÜLLER M., HENNIX B. H. M., RATCLIFF J.: Position based dynamics. *Proceedings of Virtual Reality Interactions and Physical Simulations* (2006), 71–80. 5

[MHTG05] MÜLLER M., HEIDELBERGER B., TESCHNER M., GROSS M.: Meshless deformations based on shape matching. *ACM Trans. Graph. 24*, 3 (July 2005), 471–478. URL: https://doi.org/10.1145/1073204.1073216, doi:10.1145/1073204.1073216. 2, 3, 6

[MKN*04] MÜLLER M., KEISER R., NEALEN A., PAULY M., GROSS M., ALEXA M.: Point based animation of elastic, plastic and melting objects. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Goslar, DEU, 2004), SCA '04, Eurographics Association, p. 141–151. URL: https://doi.org/10.1145/1028523.1028542, doi:10.1145/1028523.1028542. 3

[MM21] MACKLIN M., MÜLLER M.: A constraint-based formulation of stable neo-hookean materials. In *Motion, Interaction and Games* (New York, NY, USA, 2021), MIG '21, Association for Computing Machinery. URL: https://doi.org/10.1145/3487983.3488289, doi:10.1145/3487983.3488289. 2

[MMC16] MACKLIN M., MÜLLER M., CHENTANEZ N.: Xpbd: Position-based simulation of compliant constrained dynamics. In *Proceedings of the 9th International Conference on Motion in Games* (New York, NY, USA, 2016), MIG '16, Association for Computing Machinery, p. 49–54. URL: https://doi.org/10.1145/2994258.2994272, doi:10.1145/2994258.2994272. 2

[MSL*19] MACKLIN M., STOREY K., LU M., TERDIMAN P., CHENTANEZ N., JESCHKE S., MÜLLER M.: Small steps in physics simulation. In *Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2019), SCA '19, Association for Computing Machinery. URL: https://doi.org/10.1145/3309486.3340247, doi:10.1145/3309486.3340247. 4

[NMK*06] NEALEN A., MÜLLER M., KEISER R., BOXERMAN E., CARLSON M.: Physically based deformable models in computer graphics. *Computer Graphics Forum 25*, 4 (2006), 809–836. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2006.01000.x, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2006.01000.x, doi:https://doi.org/10.1111/j.1467-8659.2006.01000.x. 2

[SGK18] SMITH B., GOES F. D., KIM T.: Stable neo-hookean flesh simulation. *ACM Trans. Graph. 37*, 2 (mar 2018). URL: https://doi.org/10.1145/3180491, doi:10.1145/3180491. 2, 3

[WCL*20] WOLPER J., CHEN Y., LI M., FANG Y., QU Z., LU J., CHENG M., JIANG C.: Anisompm: Animating anisotropic damage mechanics. *ACM Trans. Graph. 39*, 4 (jul 2020). URL: https://doi.org/10.1145/3386569.3392428, doi:10.1145/3386569.3392428. 2

[WDG*19] WANG S., DING M., GAST T. F., ZHU L., GAGNIERE S., JIANG C., TERAN J. M.: Simulation and visualization of ductile fracture with the material point method. *Proc. ACM Comput. Graph. Interact. Tech. 2*, 2 (jul 2019). URL: https://doi.org/10.1145/3340259, doi:10.1145/3340259. 2

[WJST15] WANG Y., JIANG C., SCHROEDER C., TERAN J.: An adaptive virtual node algorithm with robust mesh cutting. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Goslar, DEU, 2015), SCA '14, Eurographics Association, p. 77–85. 2

[ZB05] ZHU Y., BRIDSON R.: Animating sand as a fluid. In *ACM SIGGRAPH 2005 Papers* (New York, NY, USA, 2005), SIGGRAPH '05, Association for Computing Machinery, p. 965–972. URL: https://doi.org/10.1145/1186822.1073298, doi:10.1145/1186822.1073298. 4