

High fidelity simulation of corotational linear FEM for incompressible materials

Mihai Frâncu
University of Copenhagen

Arni Asgeirsson
University of Copenhagen

Kenny Erleben
University of Copenhagen

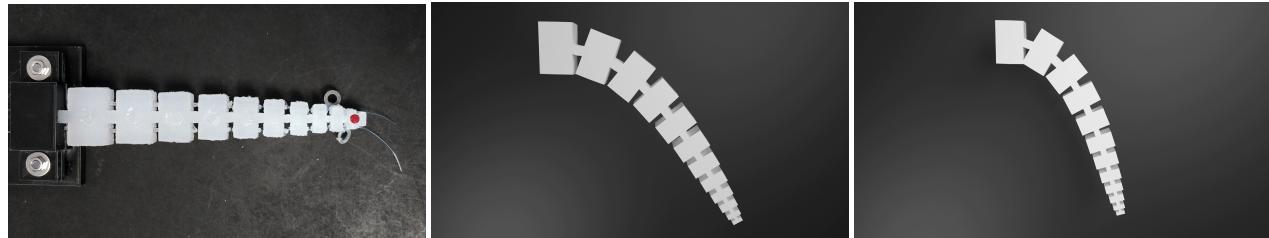


Figure 1: Soft robot made out of silicone rubber. Left: real life soft robot. Middle: the soft robot hanging under gravity simulated using corotational FEM. Right: the same simulation using our method. The corotational method suffers from locking, while our method has a wider range of motion using the same linear mesh. Elastic parameters: $E=262$ kPa, $\nu=0.49$.

ABSTRACT

We present a novel method of simulating incompressible materials undergoing large deformation without locking artifacts. We apply it for simulating silicone soft robots with a Poisson ratio close to 0.5. The new approach is based on the mixed finite element method (FEM) using a pressure-displacement formulation; the deviatoric deformation is still handled in a traditional fashion. We support large deformations without volume increase using the corotational formulation of linear elasticity. Stability is ensured by an implicit integration scheme which always reduces to a sparse linear system. For even more deformation accuracy we support higher order simulation through the use of Bernstein-Bézier polynomials.

CCS CONCEPTS

- Computing methodologies → Physical simulation.

KEYWORDS

mixed finite element, corotational elasticity, incompressibility

ACM Reference Format:

Mihai Frâncu, Arni Asgeirsson, and Kenny Erleben. 2019. High fidelity simulation of corotational linear FEM for incompressible materials. In *Motion, Interaction and Games (MIG '19), October 28–30, 2019, Newcastle upon Tyne, United Kingdom*. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3359566.3360079>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MIG '19, October 28–30, 2019, Newcastle upon Tyne, United Kingdom

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6994-7/19/10...\$15.00

<https://doi.org/10.1145/3359566.3360079>

1 INTRODUCTION

Incompressible materials form an important class of materials that has been seeing increased interest in the simulation community. In computer graphics the focus is on simulating realistic skin deformation. Other applications include biomedical ones and soft robotics. We are particularly interested in simulating and actuating such soft robots made out of silicone rubber.

Training deep neural networks for data-driven control policies in robotics requires simulations to be used a great number of times [Chebotar et al. 2018]. Therefore, the simulation has to be fast while still being as highly accurate as possible so the gap between simulation and reality is not too wide. We call these constraints the *high fidelity* requirement.

As shown by [Smith et al. 2018], incompressible rubber-like materials have been a challenge over the years. Soft robots made out of silicone rubber need to withstand large deformations and they are very stiff in terms of resisting volume change. In addition, pure displacement FEM simulation of incompressible materials is especially prone to *locking*, a numerical artifact resulting in too small deformations. This is alleviated by the *mixed* pressure-displacement formulation of FEM [Zienkiewicz et al. 2005] - see Figures 1 and 2.

In this paper we are introducing a mixed FEM method that is still based on linear elasticity, but can handle large deformations through the popular *corotational* model [Müller et al. 2008]. Higher order FEM methods have not been used a lot in computer graphics, but they hold good promise for obtaining more deformation out of low-resolution meshes at a lower memory cost. Our method allows for higher order simulation in both displacements and pressures using Bernstein-Bézier shape functions.

To summarize, the contributions made in this paper include: (i). a pressure-displacement mixed formulation of the same type as the one encountered in constrained dynamics; (ii). a linear corotational FEM implicit integration scheme that allows for large deformations without volume increase; (iii). second order displacement elements using Bernstein-Bézier polynomials mixed with linear pressure elements.

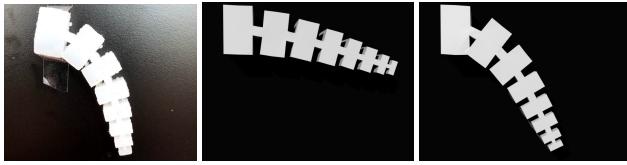


Figure 2: Smaller version of the silicone soft robot from Figure 1 hanging under gravity: left - reality, middle - corotational FEM (locking), right - our method. Dimensions: 9.5x2.5x1.5 cm; elastic parameters: E=262 kPa, $\nu=0.499$.

2 RELATED WORK

The simulation of deformable bodies in computer graphics dates back to the seminal work of [Terzopoulos et al. 1987]. We are following in the footsteps of [Roth 2002] who used a mixed pressure formulation for simulating skin. The fact that mixed formulations are very well suited for very stiff or nearly incompressible materials has been well established in the FEM literature [Zienkiewicz et al. 2005]. Another way to handle incompressibility is through a divergence free solver [Irving et al. 2007].

Mixed FEM is closely related to the *constraint-based* FEM approach first introduced by [Servin et al. 2006]. The *constrained dynamics* framework has mostly been used for simulation of rigid bodies with contact [Bender et al. 2014]. Cloth simulation is often implemented by implicit integration of mass-spring systems. This was pioneered by [Baraff and Witkin 1998] who hinted at using constraints, paving the way for *position based dynamics* (PBD) popularized by [Müller et al. 2007]. In fact PBD can be used to simulate deformable solids in many ways, summarized in [Bender et al. 2014]. A related approach to constrained dynamics and PBD is the Projective Dynamics method for simulating deformable bodies [Bouaziz et al. 2014]. More recent work on constraint-based FEM used PBD in order to enforce nonlinear material models through constraints: [Frâncu and Moldoveanu 2017; Macklin et al. 2019, 2016]. The constrained dynamics work in [Tournier et al. 2015] bears resemblance to ours through the addition of the geometric stiffness matrix.

Traditional FEM approaches (referred to as *pure displacement* methods) are ubiquitous in engineering and computer graphics. Offline simulation for movies can afford complicated nonlinear simulators with larger computations times [Smith et al. 2018]. For interactive and real-time simulation the corotational model has been the method of choice [Parker and O'Brien 2009]. The method has seen many improvements regarding element inversion [Irving et al. 2004; Schmedding and Teschner 2008] and polar decomposition [Civit-Flores and Susín 2014; Kugelstadt et al. 2018]. Regarding higher order simulation, we were inspired by [Weber et al. 2011, 2015] and [Bargteil and Cohen 2014] in choosing Bernstein-Bézier polynomials as shape functions.

3 THE MIXED FINITE ELEMENT METHOD

In this paper we are using the mixed pressure-displacement formulation of FEM for linear elasticity found in [Roth 2002] and [Zienkiewicz et al. 2005]. This approach handles the deviatoric deformations in a traditional FEM manner, whereas the volume change is treated in a mixed formulation. The volumetric stress is

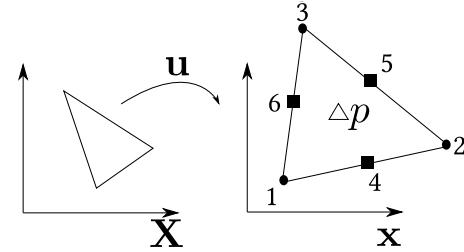


Figure 3: A 2D schematic of material and spatial coordinate systems and of a triangle element.

described by a single scalar in all normal directions, i.e. the hydrostatic pressure p .

3.1 Constrained dynamics

Given a set of constraints $\phi(q) = 0$ the dynamics of a finite dimensional mechanical system is described by the following set of *differential algebraic equations* (DAE):

$$M\ddot{q} + J^T \lambda = f, \quad (1a)$$

$$\phi(q) - C\lambda = 0, \quad (1b)$$

where M is the mass matrix, q are the generalized coordinates, $J = \nabla\phi(q)$ is the Jacobian matrix of the constraint function ϕ , λ are the Lagrange multipliers enforcing the constraints and f are the external forces acting on the system.

The extra term C in (1b) on top of standard constrained dynamics DAE is a regularization term. Its role is to soften the hard constraints and it can be shown that it is equivalent to turning the constrained system into an elastic one governed by the compliance matrix C (i.e. the inverse of stiffness). A zero compliance matrix corresponds to the limit case of infinite stiffness.

3.2 Mixed pressure-displacement formulation

In order to use concepts from continuum mechanics, we introduce material and spatial coordinates for the deformable body - see Figure 3. The former are known as reference or undeformed positions of the body and are denoted by X . The spatial coordinates x are then those that represent the deformed positions of the body. The small Cauchy strain tensor is defined as:

$$\epsilon = \frac{1}{2}(\nabla u + \nabla u^T), \quad (2)$$

where $u = x - X$ is the displacement mapping and $\nabla u = \partial u / \partial X$ is its gradient. The strain can be split into two parts: volumetric and deviatoric. The former is defined as $\epsilon_v = \text{tr}(\epsilon)$ and the deviatoric strain is obtained by subtracting the average diagonal value:

$$\epsilon_d = \epsilon - (\epsilon_v/3)I_3, \quad (3)$$

where I_3 is the 3-by-3 identity matrix. Similarly, the stress can be split into a deviatoric and a volumetric part: $\sigma = \sigma_d + pI_3$. The stress-strain relation can then be stated separately for each of the two stress and strain components:

$$p = K\epsilon_v \text{ and } \sigma_d = 2G\epsilon_d, \quad (4)$$

where K is the *bulk modulus* and G is the *shear modulus*. The relation between the bulk and the shear modulus to the Young's modulus E

and the Poisson ratio ν is given by:

$$G = E/[2(1 + \nu)], K = E/[3(1 - 2\nu)]. \quad (5)$$

It becomes clear from the second relation that for $\nu = 0.5$ the bulk modulus K becomes infinity.

We are using tetrahedral elements on which we sample both the displacement and the pressure fields (i.e. in a mixed form) using separate shape functions: $\mathbf{u} \approx \sum_{A=1}^{n_u} \mathbf{N}_u^A \tilde{\mathbf{u}}^A$ and $p \approx \sum_{B=1}^{n_p} \mathbf{N}_p^B \tilde{p}^B$, where the super-scripts A and B denote values at displacement and pressure nodes respectively; the tilde marks nodal values.

The strain inside an element can be expressed in Voigt notation:

$$\hat{\boldsymbol{\epsilon}} = \mathbf{B} \tilde{\mathbf{u}} = \mathbf{S} \mathbf{N}_u \tilde{\mathbf{u}}, \quad (6)$$

where \mathbf{N}_u and $\tilde{\mathbf{u}}$ are the aggregated displacement shape function matrix and displacement vector for the entire element, \mathbf{S} is a matrix differential operator corresponding in tensor notation to the symmetric gradient operator from (2), and $\mathbf{B} = \partial \hat{\boldsymbol{\epsilon}} / \partial \tilde{\mathbf{u}}$ is the Jacobian matrix of the strain with respect to the nodal displacements.

In order to handle the deviatoric part in a traditional fashion we need to evaluate the deviatoric stiffness matrix:

$$\mathbf{K}_d = \int_T \mathbf{B}_d^T \mathbf{E}_d \mathbf{B}_d d\mathbf{X}, \quad (7)$$

where T denotes the current tetrahedron and

$$\mathbf{E}_d = \text{diag}(2G, 2G, 2G, G, G, G). \quad (8)$$

The matrix \mathbf{B}_d is the Jacobian matrix corresponding to the deviatoric strain. It can be obtained from \mathbf{B} by the projection:

$$\mathbf{B}_d = \begin{bmatrix} \mathbf{P} & 0 \\ 0 & \mathbf{I}_3 \end{bmatrix} \mathbf{B}, \quad (9)$$

where the projection matrix \mathbf{P} is

$$\mathbf{P} = \frac{1}{3} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}. \quad (10)$$

The volumetric strain Jacobian matrix is:

$$\mathbf{B}_v = [1 \ 1 \ 1 \ 0 \ 0 \ 0] \mathbf{B}. \quad (11)$$

The aim of the pressure-displacement mixed form is to enforce the volumetric constitutive law in (4) as a constraint. For this we need to invert the law as a strain-stress relation: $\boldsymbol{\epsilon}_v - \frac{1}{K} p = 0$. In discretized form this corresponds to the following finite dimensional linear constraint function:

$$\phi(\mathbf{u}) = \int_T \mathbf{N}_p \boldsymbol{\epsilon}_v d\mathbf{X} = \mathbf{J} \tilde{\mathbf{u}}, \quad (12)$$

together with the volumetric compliance matrix:

$$\mathbf{C} = \frac{1}{K} \int_T \mathbf{N}_p^T \mathbf{N}_p d\mathbf{X}. \quad (13)$$

The constant Jacobian matrix of the constraint function in (12) is:

$$\mathbf{J} = \int_T \mathbf{N}_p \mathbf{B}_v d\mathbf{X}. \quad (14)$$

The resulting elastodynamics equations for a single element are:

$$\mathbf{M} \ddot{\mathbf{u}} + \mathbf{K}_d \tilde{\mathbf{u}} + \mathbf{J}^T \tilde{\mathbf{p}} = \tilde{\mathbf{f}}, \quad (15)$$

$$\mathbf{J} \tilde{\mathbf{u}} - \mathbf{C} \tilde{\mathbf{p}} = 0, \quad (16)$$

where \mathbf{M} is the mass matrix computed as in [Erleben et al. 2005]:

$$\mathbf{M} = \rho \int_T \mathbf{N}_u^T \mathbf{N}_u d\mathbf{X}, \quad (17)$$

with ρ being the density of the material.

After assembling all the matrices and vectors for all the elements we get exactly the constrained dynamics DAE in (1a)-(1b) with the \mathbf{q} representing the displacements vector and λ the pressures vector.

3.3 Bernstein-Bézier shape functions

In order to give more detailed formulas we need to introduce some index notation. The constraint function ϕ in (12) is a mapping from the n_u displacement nodes to the n_p pressure nodes. Its Jacobian matrix \mathbf{J} has n_p rows and $3n_u$ columns. For a displacement node of index $I = 1..n_u$ the Jacobian matrix \mathbf{B} of the strain is:

$$B_{mn}^I = B_{(ij)n}^I = \frac{\partial \epsilon_{ij}}{\partial \tilde{u}_n^I}, \quad (18)$$

where $i, j, n = 1..3$ and $m = 1..6$. The $m \equiv (ij)$ indices mark the equivalence between Voigt and tensor notation.

For computing the Jacobian matrix \mathbf{J} of the constraint function one has to first evaluate \mathbf{B}_v from (11) using (18). After choosing the shape functions one can use the formula in (14) to compute the Jacobian block matrix corresponding to displacement node I and the pressure node $A = 1..n_p$:

$$\mathbf{J}^{AI} = \int_T \mathbf{N}_p^A \mathbf{B}_v^I d\mathbf{X}. \quad (19)$$

Bernstein-Bézier polynomials of degree q have the form:

$$\mathcal{B}_{ijkl}^q(\alpha) = \frac{q!}{i!j!k!l!} \alpha_1^i \alpha_2^j \alpha_3^k \alpha_4^l, \quad (20)$$

where α are the four barycentric coordinates inside the tetrahedron. The indices $ijkl$ form a multi-index I that can also be seen as a local node index going from 1 to n_u (or n_p). The main property of this index set is that $|I| = i + j + k + l = q$.

For linear displacement and constant pressure elements ($N_p \equiv 1$), with V_T the undeformed tetrahedron volume, we have:

$$\mathbf{K}_d^{IJ} = \int_T (\mathbf{B}_d^I)^T \mathbf{E}_d \mathbf{B}_d^J d\mathbf{X}, \quad (21)$$

$$\mathbf{J}^{AI} = \int_T \mathbf{B}_v^I d\mathbf{X} = V_T \mathbf{B}_v^I, \quad (22)$$

$$\mathbf{C}^{AI} = \frac{1}{K} \int_T \mathbf{I}_{n_p} d\mathbf{X} = \frac{V_T}{K} \mathbf{I}_{n_p}. \quad (23)$$

We can bump up the order of the pressure elements from piece-wise constant per element to first order and arrive at new formulas:

$$\mathbf{J}^{AI} = \int_T \alpha_A \mathbf{B}_v^I d\mathbf{X} = \frac{V_T}{4} \mathbf{B}_v^I, \quad (24)$$

$$\mathbf{C}^{AB} = \frac{1}{K} \int_T \alpha_A \alpha_B d\mathbf{X} = \frac{V_T}{10K} \mathcal{G}^1(A, B), \quad (25)$$

where the \mathcal{G}^q function used for multiplying two Bernstein-Bézier polynomials is given in Section 3 of [Weber et al. 2011].

For the case of quadratic displacements, we chose as in [Roth 2002] first order pressure elements. For a description of quadratic elements one can see a 2D analogy in Figure 3. The circle corner nodes represent the nodes that are also present in a linear element. The mid-edge square nodes are the extra nodes of the second order

mesh. For linear pressure elements the values are located at the nodes. The strain Jacobian computed from (18) becomes a little more complicated (and not constant anymore over the element):

$$B_{(ij)k}^I(\boldsymbol{\alpha}) = \sum_{n=1}^4 \mathcal{B}_{I-[n]}^1(\boldsymbol{\alpha})(y_j^n \delta_{ik} + y_i^n \delta_{jk}), \quad (26)$$

where δ_{ij} is the Kronecker delta symbol and the notation $I - [n]$ was introduced by [Bargteil and Cohen 2014] – it represents the multi-index I from which the n th index is decremented by one. The barycentric coordinates derivative $\partial \alpha_n / \partial X_i$ is denoted by y_i^n . The constraint Jacobian matrix has then the following form:

$$J_{(ij)k}^{AI} = \frac{1}{10} \sum_{n=1}^4 \mathcal{G}^1(A, I - [n])(y_j^n \delta_{ik} + y_i^n \delta_{jk}). \quad (27)$$

3.4 Implicit incompressible corotational

For the time dependent problem the FEM discretization provides us with a DAE that requires numerical time integration and constraint solving. In this section we are deriving equations for one element only that can later be assembled into a global system. The volumetric constraint in (16) is always treated in an implicit fashion, as we require it to be satisfied at the end of the time-step:

$$\tilde{\mathbf{J}}\tilde{\mathbf{u}}^{t+1} - \mathbf{C}\tilde{\mathbf{p}}^{t+1} = 0. \quad (28)$$

Note that the constraint Jacobian matrix \mathbf{J} is constant throughout the whole simulation. We can decide whether to integrate the deviatoric forces in an explicit or implicit manner.

When unconditional stability of the simulation is required, we choose to use implicit Euler integration. The derivative of the deviatoric elastic force \mathbf{K}_d is constant at all times, hence we do not need to employ a non-linear solver. The system is always linear and has a constant system matrix:

$$\begin{bmatrix} \mathbf{M} + h^2 \mathbf{K}_d & h\mathbf{J}^T \\ h\mathbf{J} & -\mathbf{C} \end{bmatrix} \begin{pmatrix} \tilde{\mathbf{v}}^{t+1} \\ \tilde{\mathbf{p}}^{t+1} \end{pmatrix} = \begin{pmatrix} \mathbf{M}\tilde{\mathbf{v}}^t - h\mathbf{K}_d\mathbf{u}^t + h\mathbf{f}^t \\ -\mathbf{J}\mathbf{u}^t \end{pmatrix}. \quad (29)$$

The system (29) works very well for small deformations as all of the matrices remain constant and can be pre-computed. However, this is unusable for large deformations as there will be volume increase under rotations due to the linear model. Therefore, we employ a linear corotational approach to our implicit integration scheme in (29).

Given the deformation gradient $\mathbf{F} = \partial \mathbf{x} / \partial \mathbf{X} = \nabla \mathbf{u} - \mathbf{I}$ we can extract an orthonormal rotation matrix \mathbf{R} so that $\mathbf{F} = \mathbf{R}\mathbf{U}$. For computing \mathbf{R} one can use polar decomposition (resulting in symmetric \mathbf{U}) or the Gram-Schmidt QR factorization from [Müller et al. 2008]. We now express the elastic force as:

$$\mathbf{f}_{cr} = \mathbf{R}\mathbf{K}_d(\mathbf{R}^T \tilde{\mathbf{x}} - \mathbf{X}) = \mathbf{R}\mathbf{K}_d\tilde{\mathbf{u}}_{cr}. \quad (30)$$

Using this force expression, the final equations of the implicit incompressible corotational (IICR) mixed form FEM are:

$$\begin{bmatrix} \mathbf{M} + h^2 \mathbf{R}^T \mathbf{K}_d \mathbf{R} & h\mathbf{R}\mathbf{J}^T \\ h\mathbf{J}\mathbf{R}^T & -\mathbf{C} \end{bmatrix} \begin{pmatrix} \tilde{\mathbf{v}}^{t+1} \\ \tilde{\mathbf{p}}^{t+1} \end{pmatrix} = \begin{pmatrix} \mathbf{M}\tilde{\mathbf{v}}^t - h\mathbf{f}_{cr}^t + h\mathbf{f}^t \\ -\mathbf{J}\tilde{\mathbf{u}}_{cr}^t \end{pmatrix}. \quad (31)$$

4 IMPLEMENTATION DETAILS

We have implemented our code in C++ and used the Eigen library [Guennebaud et al. 2010] for linear algebra. Most of the time we

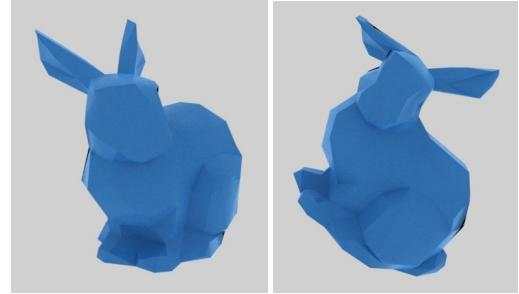


Figure 4: Bunny hanging from one ear simulated using IICR (left - initial configuration): there are no volume artifacts.

used sparse direct solvers based on LDLT decomposition (as recommended in [Benzi et al. 2005]) or LU decomposition for KKT form systems. Calculations were done mostly in single precision as they are faster and the memory footprint is smaller.

All of our simulations use a gravitational acceleration of 9.8 m/s^2 and elastic parameters specific to silicone rubber: a density between 1070 and 1250 kg/m^3 , a Young's modulus between 66 and 2500 kPa and a Poisson ratio of 0.49 or more. The time-step of the simulation is set to a reference step of 16 ms (corresponding to rendering at 60 Hz) and can be subdivided into a number of sub-steps.

The tetrahedral meshes were generated using NetGen [Schöberl 2009]. We have used TetGen [Si 2015] for our cantilever experiments. We computed the mass matrix both as stated in equation (17) and using the lumped mass approximation [Erleben et al. 2005].

The assembly process for the corotated stiffness matrix and Jacobians in (31) becomes slightly more complicated, as the rotation matrix corresponds to one element at any given moment of time. Just like [Weber et al. 2011] we are using the same rotation matrix for quadratic elements too. The system in (31) is a sparse system known as a saddle point problem. Solving such systems can be challenging as the system matrix is either symmetric or positive definite, but not both at the same time. We are choosing not to use a Schur complement approach in this paper. By exploiting the sparsity of the KKT system matrix we were able to obtain better performance than with dense solvers. To speed up performance we have used the Intel Math Kernel Library (MKL) [Wang et al. 2014] support in Eigen together with its PARDISO [Schenk and Gärtner 2004] solver integration.

On the iterative solvers side we have tried Conjugate Gradient (CG), MINRES and GMRES for the KKT system. For GMRES using ILUT (incomplete LU) preconditioning worked best. For CG and MINRES robust solutions were obtained only by using the incomplete Cholesky preconditioner. Limiting the number of iterations always resulted in instability.

5 RESULTS

One experiment we did was to hang the Stanford bunny by one ear and let it swing under gravity - see Figure 4. For linear elasticity this manifests into very high distortion due to volume increase. Our method (IICR) and the traditional corotational method (CR) behave the same and preserve the volume - see Figure 5. For constant pressure elements (IICR10) we noticed almost no difference at all to

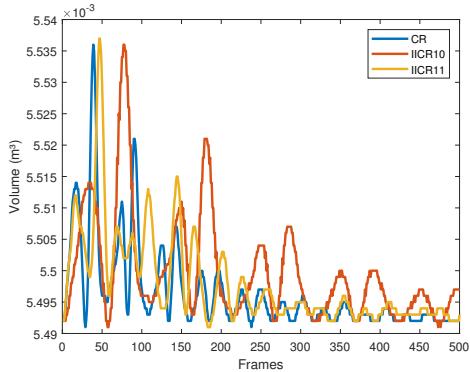


Figure 5: Plot of the total volume of bunny mesh over 500 frames. Our method (IICR10 and IICR11) preserves volume around the same values as corotational (CR).

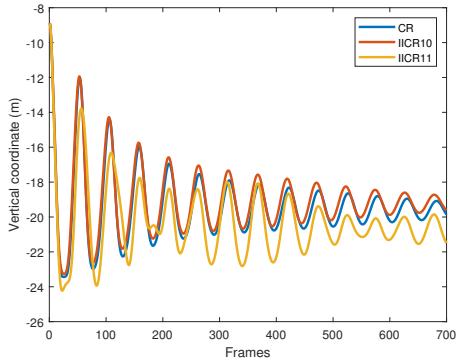


Figure 6: Plot of vertical coordinate of a point in the tail of the bunny. Our method (IICR10 and IICR11) is very similar to corotational (CR) with only IICR11 going out of sync.

Table 1: Comparison of CPU times (ms) per frame between corotational (CR) and our method (IICR11 and IICR10). Our method is only slightly slower when solving a larger system.

Model	#tets	#nodes	CR	IICR11	Increase	IICR10
Bunny	749	280	8	10	25%	15
Dragon	834	343	11	14	27%	18
Cow	972	334	11	15	36%	20
Spine	1778	649	40	43	7%	69
TorusLow	1934	513	27	39	44%	60
Hinge	2429	826	70	75	7%	122
TorusHigh	2823	908	83	106	28%	162

corotational FEM, whereas for linear pressure elements (IICR11) the movement stops overlapping after a number of frames - see Figure 6.

Locking manifests in displacements that are much smaller than the real ones. In the traditional corotational method one can notice that for very high Poisson ratios and low resolution meshes the deformation is quite small and unrealistic - see Figure 7. Even our

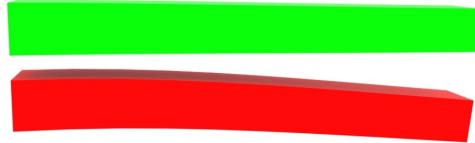


Figure 7: Locking phenomenon for the green cantilever beam (CR) for large Poisson ratios. This is fixed for the red one using our method (IICR11). Cantilever details: 10x1x1 cm (192 tetrahedra), E=262 kPa, $\nu=0.49$.

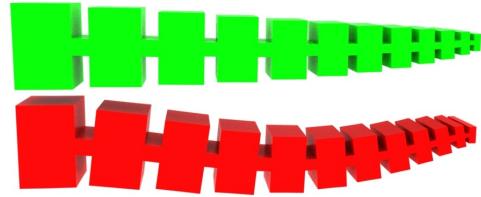


Figure 8: The large soft robot hanging from its two ends under gravity. The green one (CR) completely locks, whereas the red one (IICR11) deforms as expected even for very large Poisson ratios. Elastic parameters: E=262 kPa, $\nu=0.49999$.

mixed FEM method using constant pressure elements can yield such bad results for linear elements (IICR10). The remedy comes from using linear pressure elements (IICR11).

We have tested our method on the "spine" soft robot model in Figure 1 and obtained a larger domain of movement than for corotational FEM. We have tested it on a smaller version of the robot and increased the Poisson ratio further and noticed an even bigger difference. You can see in Figure 2 that we closely match reality, although we are not modeling self collisions. We were not able to do the same experiment for the larger robot due to the actuation cables. In Figure 8, we hung our large simulated soft robot by its two ends and increased the Poisson ratio even more. The result was an almost complete locking of the corotational simulation while our method looked roughly the same for all Poisson ratio values and more realistic.

For analyzing performance we have gathered timings on a desktop computer with an Intel Xeon W-2133 CPU. You can find a comparison between our method (IICR) and our implementation of corotational FEM in Table 1 (values were rounded). The two methods have been optimized as much as possible and the fastest solver for each of them was chosen. As expected, the linear pressure elements version (IICR11) is faster than the piece-wise constant one (IICR10) as there are less unknowns in the system (the decrease is from the number of elements to the number of nodes).

In Figure 9 one can see a comparison using second order meshes between our method and traditional corotational FEM. Our second order method is using linear pressure elements and is denoted by IICR21. As it can be seen there is no significant difference between the bending of the cantilever. This is because quadratic corotational FEM handles locking much better. On the other hand, the linear mesh simulation has a hard time converging to the correct solution

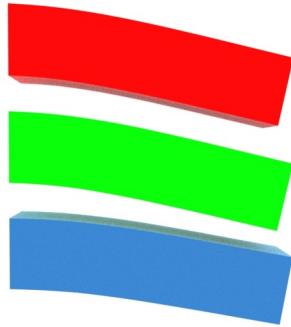


Figure 9: Second order simulation of cantilever beam with 192 tetrahedra: our method (IICR21) in red, corotational (CR) in green. Both second order methods behave just as good compared to a very high resolution linear mesh with 24000 tetrahedra (in blue) even for low Poisson ratios. Cantilever details: 4x1x1 cm, E=66 kPa, $\nu=0.25$.

even using a very high resolution and a relatively small Poisson ratio. In conclusion, there is a clear advantage in using quadratic elements, as outlined in [Weber et al. 2011], for obtaining more deformation out of low resolution meshes.

6 CONCLUSIONS

We have introduced a new FEM method for simulating elastic incompressible materials: IICR. It is a little more expensive than the traditional corotational method, but it handles locking much better for high Poisson ratios. For large systems with many degrees of freedom the GPU will most likely be the only viable solution for obtaining interactive performance. Collisions can be implemented in the future through penalty forces or by using a mixed LCP solver for the KKT system.

ACKNOWLEDGMENTS

The first author has received funding from the European Union's Horizon 2020 research and innovation programme under Marie Skłodowska-Curie grant agreement No. 799001. This report only contains the authors views and the Research Executive Agency and the Commission are not responsible for any use that may be made of the information it contains.

We would like to thank Max Kragballe Nielsen for providing us the silicone soft robots. Daniel Weber helped us with validating our second order method. Rendering was done with the help of Alin Dumitru and Liviu Dinu from StaticVFX studio.

REFERENCES

- David Baraff and Andrew Witkin. 1998. Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM, 43–54.
- Adam W Bargteil and Elaine Cohen. 2014. Animation of deformable bodies with quadratic Bézier finite elements. *ACM Transactions on Graphics (TOG)* 33, 3 (2014), 27.
- Jan Bender, Kenny Erleben, and Jeff Trinkle. 2014. Interactive simulation of rigid body dynamics in computer graphics. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 246–270.
- Michele Benzi, Gene H Golub, and Jörg Liesen. 2005. Numerical solution of saddle point problems. *Acta numerica* 14 (2005), 1–137.
- Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. 2014. Projective dynamics: fusing constraint projections for fast simulation. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 154.
- Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. 2018. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. *arXiv preprint arXiv:1810.05687* (2018).
- Oscar Civit-Flores and Antonio Süsli. 2014. Robust treatment of degenerate elements in interactive corotational FEM simulations. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 298–309.
- Kenny Erleben, Jon Sporring, Knud Henriksen, and Henrik Dohlmann. 2005. *Physics-based animation*. Charles River Media.
- Mihai Frâncu and Florica Moldoveanu. 2017. Position based simulation of solids with accurate contact handling. *Computers & Graphics* 69 (2017), 12–23.
- Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen v3. <http://eigen.tuxfamily.org>.
- Geoffrey Irving, Craig Schroeder, and Ronald Fedkiw. 2007. Volume conserving finite element simulations of deformable models. In *ACM Transactions on Graphics (TOG)*, Vol. 26. ACM, 13.
- Geoffrey Irving, Joseph Teran, and Ronald Fedkiw. 2004. Invertible finite elements for robust simulation of large deformation. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 131–140.
- Tassilo Kugelstadt, Dan Koschier, and Jan Bender. 2018. Fast Corotated FEM using Operator Splitting. *Computer Graphics Forum (SCA)* 37, 8 (2018).
- Miles Macklin, Kenny Erleben, Matthias Müller, Nuttapong Chentanez, Stefan Jeschke, and Viktor Makoviychuk. 2019. Non-Smooth Newton Methods for Deformable Multi-Body Dynamics. *arXiv preprint arXiv:1907.04587* (2019).
- Miles Macklin, Matthias Müller, and Nuttapong Chentanez. 2016. XPBD: position-based simulation of compliant constrained dynamics. In *Proceedings of the 9th International Conference on Motion in Games*. ACM, 49–54.
- Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. 2007. Position based dynamics. *Journal of Visual Communication and Image Representation* 18, 2 (2007), 109–118.
- Matthias Müller, Jos Stam, Doug James, and Nils Thuerey. 2008. Real time physics: class notes. In *ACM SIGGRAPH 2008 classes*. ACM, 88.
- Eric G Parker and James F O'Brien. 2009. Real-time deformation and fracture in a game environment. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 165–175.
- Samuel Hans Martin Roth. 2002. *Bernstein-Bézier representations for facial surgery simulation*. Vol. 16. ETH Zurich.
- Olaf Schenk and Klaus Gärtner. 2004. Solving unsymmetric sparse systems of linear equations with PARDISO. *Future Generation Computer Systems* 20, 3 (2004), 475–487.
- Ruediger Schmedding and Matthias Teschner. 2008. Inversion handling for stable deformable modeling. *The Visual Computer* 24, 7–9 (2008), 625–633.
- Joachim Schöberl. 2009. NETGEN-4. X. *RWTH Aachen University, Germany* (2009).
- Martin Servin, Claude Lacourciere, and Niklas Melin. 2006. Interactive simulation of elastic deformable materials. In *SIGRAD 2006. The Annual SIGRAD Conference; Special Theme: Computer Games*. Linköping University Electronic Press.
- Hang Si. 2015. TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator. *ACM Trans. Math. Softw.* 41, 2, Article 11 (Feb. 2015), 36 pages.
- Breannan Smith, Fernando De Goes, and Theodore Kim. 2018. Stable neo-hookean flesh simulation. *ACM Transactions on Graphics (TOG)* 37, 2 (2018), 12.
- Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. 1987. Elastically deformable models. *ACM Siggraph Computer Graphics* 21, 4 (1987), 205–214.
- Maxime Tournier, Matthieu Nesme, Benjamin Gilles, and François Faure. 2015. Stable constrained dynamics. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 132.
- Endong Wang, Qing Zhang, Bo Shen, Guangyong Zhang, Xiaowei Lu, Qing Wu, and Yajuan Wang. 2014. Intel math kernel library. In *High-Performance Computing on the Intel® Xeon Phi*. Springer, 167–188.
- Daniel Weber, Thomas Kalbe, André Stork, Dieter Fellner, and Michael Goesele. 2011. Interactive deformable models with quadratic bases in Bernstein-Bézier-form. *The visual computer* 27, 6–8 (2011), 473.
- Daniel Weber, Johannes Mueller-Roemer, Christian Altenhofen, André Stork, and Dieter Fellner. 2015. Deformation simulation using cubic finite elements and efficient p-multigrid methods. *Computers & graphics* 53 (2015), 185–195.
- Olek C Zienkiewicz, Robert L Taylor, and Jian Z Zhu. 2005. *The finite element method: its basis and fundamentals*. Elsevier.

