High fidelity simulation of corotational linear FEM for incompressible materials

Francu, Mihail; Asgeirsson, Arni; Erleben, Kenny

Published in:
Proceedings - MIG 2019

DOI:
10.1145/3359566.3360079

Publication date:
2019

Document version
Peer reviewed version

Citation for published version (APA):
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

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.

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...
where \( M = 3.1 \) Constrained dynamics external forces acting on the system. forces acting on the system.

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):

\[
\begin{align*}
M\ddot{q} + J^T\lambda &= f, \quad (1a) \\
\phi(q) - C\lambda &= 0, \quad (1b)
\end{align*}
\]

where \( M \) is the mass matrix, \( q \) are the generalized coordinates, \( J = \nabla\phi(q) \) is the Jacobian matrix of the constraint function \( \phi \), \( \lambda \) 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

We first 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:

\[
\varepsilon = \frac{1}{2}(\nabla u + \nabla u^T),
\]

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 \( \varepsilon_V = \text{tr}(\varepsilon) \) and the deviatoric strain is obtained by subtracting the average diagonal value:

\[
\varepsilon_d = \varepsilon - (\varepsilon_V / 3) I_3,
\]

where \( I_3 \) is the 3-by-3 identity matrix. Similarly, the stress can be split into a deviatoric and a volumetric part:

\[
\sigma = \sigma_d + \rho I_3.
\]

The stress-strain relation can then be stated separately for each of the two stress and strain components:

\[
p = K \varepsilon_V \quad \text{and} \quad \sigma_d = 2G \varepsilon_d,
\]

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 \( \nu \) is given by:

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

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:

\[
\nu \approx \sum_{A=1}^{n_A} N_A^\nu \hat{u}^A \quad \text{and} \quad \rho \approx \sum_{B=1}^{n_B} N_B^\rho \hat{p}^B,
\]

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:

\[
\hat{\varepsilon} = B \hat{u} = \Sigma N_B \hat{u}^B,
\]

where \( N_B \) and \( \hat{u}^B \) are the aggregated displacement shape function matrix and displacement vector for the element, \( \hat{u} \) is a matrix
differential operator corresponding in tensor notation to the symmetric gradient operator from (2), and
\[ B = \frac{\partial \mathbf{E}}{\partial \mathbf{u}} \] 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:
\[ K_d = \int_T B_d^T E_d B_d \, dX, \]  
where \( T \) denotes the current tetrahedron and
\[ E_d = \text{diag}(2G, 2G, 2G). \]  
The matrix \( B_d \) is the Jacobian matrix corresponding to the deviatoric strain. It can be obtained from \( B \) by the projection:
\[ B_d = \left[ \begin{array}{ccc} P & 0 & 0 \\ 0 & 1_3 \end{array} \right] B, \]  
where the projection matrix \( P \) is
\[ P = \frac{1}{3} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}. \]

The volumetric strain Jacobian matrix is:
\[ B_v = \left[ \begin{array}{cccc} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{array} \right] B. \]  
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:
\[ \epsilon = J^{-1} \sigma. \]  
In discretized form this corresponds to the following finite dimensional linear constraint function:
\[ \phi(u) = \int_T N_p \epsilon_v \, dX = J \tilde{u}, \]  
with the volumetric compliance matrix:
\[ C = \frac{1}{K} \int_T N_p^T N_p \, dX. \]  
The constant Jacobian matrix of the constraint function in (12) is:
\[ J = \int_T N_p B_v \, dX. \]  
The resulting elastodynamics equations for a single element are:
\[ M \ddot{u} + K_d \dot{u} + J^T \dot{\tilde{p}} = \tilde{f}, \]  
\[ J \ddot{u} - C \dot{\tilde{p}} = 0, \]  
where \( M \) is the mass matrix computed as in [Erleben et al. 2005]:
\[ M = \rho \int_T N_u^T N_u \, dX, \]  
with \( \rho \) 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 \( q \) representing the displacements vector and \( \lambda \) the pressures vector.

\[ B^l_{mn} = B^l_{(ij)n} = \frac{\partial \mathbf{E}_{ij}}{\partial u_m}, \]  
where \( i, j, n = 1..3 \) and \( m = 1..6 \). The \( m = (ij) \) indices mark the equivalence between Voigt and tensor notation.

For computing the Jacobian matrix \( J \) of the constraint function one has to first evaluate \( B_v \) 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 \( A = 1..n_p \):
\[ J^{AI} = \int_T N^{AI}_p B_v \, dX. \]  

Bernstein-Bézier polynomials of degree \( q \) have the form:
\[ B_{ijkl}^q(\alpha) = \frac{q!}{i!j!k!l!} \alpha^i_1 \alpha^j_2 \alpha^k_3 \alpha^l_4, \]  
where \( \alpha \) are the four barycentric coordinates inside the tetrahedron.

For linear displacement and constant pressure elements (\( n_p = 1 \)), with \( V_T \) the undeformed tetrahedron volume, we have:
\[ K^l_{d} = \int_T (B_d^l)^T E_d B_d^l \, dX, \]  
\[ J^{AI} = \int_T B_v^l \, dX = V_T B_v^l, \]  
\[ C^{AI} = \frac{1}{K} \int_T N_p \, dX = \frac{V_T}{K} \text{I}_{n_p}. \]  

We can bump up the order of the pressure elements from piece-wise constant per element to first order and arrive at new formulas:
\[ J^{AI} = \int_T \alpha_A B_v^l \, dX = \frac{V_T}{4} B_v^l, \]  
\[ C^{AB} = \frac{1}{K} \int_T \alpha_A \alpha_B \, dX = \frac{V_T}{10K} G^1(A, B), \]  
where the \( G^q \) 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 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):
\[ B_{(ij)k}^l(\alpha) = \sum_{n=1}^{4} B_{(ij)n}^l(\alpha)(q_n \delta_{ik} + y_n \delta_{jk}), \]  
where \( \delta_{ij} \) is the Kronecker delta symbol and the notation \( l = [n] \) 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:

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

(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:

$$J\dot{u}^{t+1} - C P^{t+1} = 0.$$  

(28)

Note that the constraint Jacobian matrix $J$ 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 $K_d$ 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:

$$\begin{bmatrix}
M + h^2 R^T K_d R & h R^T \\
\hbar J & -C
\end{bmatrix}
\begin{bmatrix}
\tilde{v}^{t+1} \\
\tilde{p}^{t+1}
\end{bmatrix}
= \begin{bmatrix}
M \tilde{v}^{t} - h K_d \tilde{u}^{t} + hf^t \\
-h J \tilde{u}^{t}
\end{bmatrix}.$$  

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

$$f_{cr} = RK_d(R^T \dot{x} - X) = RK_d \dot{\tilde{u}}_{cr}.$$  

(30)

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

$$\begin{bmatrix}
M + h^2 R^T K_d R & h R^T \\
\hbar J R^T & -C
\end{bmatrix}
\begin{bmatrix}
\tilde{v}^{t+1} \\
\tilde{p}^{t+1}
\end{bmatrix}
= \begin{bmatrix}
M \tilde{v}^{t} - h K_d \tilde{u}^{t} + hf^t \\
-h J \tilde{u}^{t}
\end{bmatrix}.$$  

(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 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 2004] solver integration. 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 the same and preserve the volume - see Figure 5. For constant 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 mixed FEM method using constant pressure elements can yield...
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.

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

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 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.
We have introduced a new FEM method for simulating elastic in-compressible 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.

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


