<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>Development of a Finite Volume Based Structural Solver for Large Rotation of Non-Orthogonal Meshes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Authors(s)</strong></td>
<td>Cardiff, Philip; Ivankovic, Alojz; Karac, Aleksandar; et al.</td>
</tr>
<tr>
<td><strong>Publication date</strong></td>
<td>2012</td>
</tr>
<tr>
<td><strong>Conference details</strong></td>
<td>7th OpenFOAM Workshop, Technische Universität Darmstadt Germany 25-28 June 2012 Darmstadt</td>
</tr>
<tr>
<td><strong>Publisher</strong></td>
<td>OpenFOAM Workshop</td>
</tr>
<tr>
<td><strong>Item record/more information</strong></td>
<td><a href="http://hdl.handle.net/10197/4768">http://hdl.handle.net/10197/4768</a></td>
</tr>
</tbody>
</table>

Downloaded 2019-11-17T11:29:35Z

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd_oa)

Some rights reserved. For more information, please see the item record link above.
Development of a Finite Volume Based Structural Solver for Large Rotation of Non-Orthogonal Meshes

Philip Cardiff

philip.cardiff@ucd.ie
philip.cardiff@gmail.com

Alojz Ivanković
Aleksandar Karač
Željko Tuković
Presentation Outline

Background & Motivation
Mathematical Model
Implementation in OpenFOAM
Numerical Test Cases
Mesh Movement
Summary & Conclusions
Background & Motivation
Stance with Active Hill Muscle

*Gluteus Medius*
Half Gait cycle

heel-strike to push-off

Time: 0.000 s
Mathematical Model
Implementation in OpenFOAM
Finite Volume Stress Analysis

**Governing Equation**

\[
\frac{\delta}{\delta t} \int_{\Omega} \rho \mathbf{v} \, d\Omega = \int_{\Gamma} \mathbf{\sigma} \cdot d\Gamma + \int_{\Omega} \rho \mathbf{b} \, d\Omega
\]

**Rate of Momentum**  **Surface Forces**  **Body Forces**

**Constitutive Relation**

\[
\mathbf{S} = 2\mu \mathbf{E} + \lambda \text{tr}(\mathbf{E}) \mathbf{I}
\]

2nd Piola-Kirchhoff Stress Tensor

**Strain Measure**

\[
\mathbf{E} = \frac{1}{2} \left( \nabla \mathbf{u} + \nabla \mathbf{u}^T + \nabla \mathbf{u}^T \cdot \nabla \mathbf{u} \right)
\]

Conservation of Linear Momentum

St. Venant-Kirchhoff Hyperelastic Relation

Lagrangian Green Strain Tensor
Mathematical Model

Updated Lagrangian Approach

\[
\frac{\delta}{\delta t} \int_{\Omega_u} \rho_u \frac{d\bar{u}}{dt} d\Omega_u = \frac{1}{\delta t} \int_{\Gamma_u} [2\mu \delta \mathbf{E}_u + \lambda tr(\delta \mathbf{E}_u) \mathbf{I}] \cdot d\Gamma_u + \\
\frac{1}{\delta t} \int_{\Gamma_u} [\mathbf{S}_u \cdot \delta \mathbf{F}_u^T + \delta \mathbf{S}_u \cdot \delta \mathbf{F}_u^T] \cdot d\Gamma_u + \int_{\Omega_u} \rho_u \frac{\delta \mathbf{b}}{\delta t} d\Omega_u
\]

Decompose into Implicit & Explicit Terms

\[
\frac{d}{dt} \int_{\Omega_\phi} \rho_\phi \frac{d\phi}{dt} d\Omega_\phi = \int_{\Gamma_\phi} (2\mu + \lambda) \nabla \phi \cdot d\Gamma_\phi + \int_{\Gamma_\phi} \mathbf{Q}_\Gamma \cdot d\Gamma_\phi + \int_{\Omega_\phi} \mathbf{Q}_\Omega d\Omega_\phi
\]
Implementation in OpenFOAM-1.6-ext

elasticNonLinSolidFoam

fvVectorMatrix DUEqn
(
    fvm::d2dt2(rho,DU)
    ==
    fvm::laplacian(2*mu + lambda, DU)
    + fvc::div
        (mu*gradDU.T() + lambda*(I*tr(gradDU))
        - (mu + lambda)*gradDU
        + mu*(gradDU & gradDU.T())
        + 0.5*lambda*(gradDU & & gradDU)*I
        + ((sigma + DSigma) & DF.T())
    );
Discretisation & Non-Orthogonal Correction

\[ \int_{\Gamma_\phi} D \left( \nabla \phi \right) \cdot d\Gamma_\phi = \sum_{f=1}^{F} \int_{\Gamma_f} D_f (\nabla \phi)_f \cdot d\Gamma_f \]

\[ \approx \sum_{f=1}^{F} D_f \left[ |\Delta_f| \frac{\phi_N - \phi_P}{|d_f|} + k_{f} \cdot (\nabla \phi)_f \right] |\Gamma_f| \]

Implementation of Custom Boundary Conditions with Non-Orthogonal Correction

**fixedDisplacement**

**solidTraction**

**solidSymmetryPlane**

**solidDirectionMixed**
Numerical Test Cases

Steady-State Rotation of a Sphere
Rotation of a Sphere

During quasi-static rotation, a body experiences zero stress. To examine that the current updated Lagrangian procedure correctly predicts zero stress, a sphere, radius = 0.1 m, subjected to a quasi-static rotation is considered. The mesh of the sphere, created in commercial meshing software ANSYS ICEM CFD [47], contains 1,664 hexahedral cells, some of which are moderately non-orthogonal, as shown in Figure 1.6. A Young's modulus of 200 GPa, and a Poisson's ratio of 0.3 has been employed.

A rotational displacement of 90° about the z-axis is applied to the boundary surface of the sphere, in increments of 1°. The displacement increment for boundary face \( f \) is given by:

\[
\delta u_f = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot C_f - C_f
\]

where \( \theta \) is the increment of rotation, and \( C_f \) is the positional vector of the boundary face centre.

As the small strain approach neglects the higher order strain terms, unphysical stresses are predicted when there are large rotations [31]. To check this, the current rotation problem has additionally been simulated using the developed small strain total Lagrangian solver. For 1° of rotation, the small strain solver predicts the uniform stress.
Rotation of a Sphere

Initial Attempt with fixedValue Boundary (no boundary non-orthogonal correction)
Rotation of a Sphere

After 1° Rotation

fixedValue

fixedDisplacement
Rotation of a Sphere

Tight solution tolerance required to ensure correct zero stresses
Rotation of a Sphere

Mesh movement becomes skewed

Standard inverse-distance interpolation

Time: 0 s
Mesh Movement
Interpolation from cell centres to vertices
Interpolation

Determine value at P from values at N

Consider an internal vertex, $P$, in an unstructured mesh, as shown in Figure 1.4. The value of variable $\phi$ is known at the $N$ neighbour cell centres, $N_i$. It is required to calculate a good approximation of $\phi$ at the vertex $P$.

We want to find the value of $\phi$ at point $P$. We know the value of $\phi$ at all neighbours $N$. $P$ is at a vertex, $N$ are at cell centres. The cells are of arbitrary shape and orientation, and $P$ is connected to an arbitrary number of cells but at least one.

We fit a plane to the values of $\phi$ at the neighbours $N$ surrounding $P$. A plane has the general equation:

$$LS(x, y, z) = ax + by + cz + d$$

We use the least squares method to determine the best fit values of $a$, $b$, $c$, and $d$. The error at $N$, $e_N$, between $N$ and the plane $LS$ is:

$$e_N = LS(x, y, z) - ax_N - by_N - cz_N - d$$

Where $x_N$, $y_N$, and $z_N$ are the 3D spatial coordinates of the point $N$. **Figure 1.4:** Internal Vertex, $P$, Surrounded by Cell Centres, $N_i$.
Inverse Distance Interpolation

\[
\phi_P = \frac{\sum_{i=1}^{N} \omega_{PN_i} \phi_{Ni}}{\sum_{i=1}^{N} \omega_{PN_i}}
\]

\[
\omega_{PN_i} = \frac{1}{|r_P - r_{Ni}|}
\]
The method of least squares is used to determine the coefficients of a general plane. A general plane may be fit through the value of the positional vector of a cell centre by finding a weighted average of the values at the surrounding cell centres,

\[ \phi(x, y, z) = ax + by + cz + d \]

where \( a, b, c, \) and \( d \) are the coefficients of the plane, and \( x, y, \) and \( z \) are the spatial coordinates. Let \( e_{NI} \) be the error at each neighbour, \( N_i \), calculated by the inverse distance method:

\[ e_{NI} = \phi(x_{NI}, y_{NI}, z_{NI}) - \phi_{NI} \]

where \( x_{NI}, y_{NI}, \) and \( z_{NI} \) are the positional vectors of vertex \( i \), and \( \phi_{NI} \) is the value of \( \phi \) at vertex \( i \). The inverse distance interpolation method essentially calculates the value of \( \phi \) at a given location by finding a weighted average of the values at the surrounding cell centres,

\[ J = \sum_{i=1}^{N} \left[ ax_{NI} + by_{NI} + cz_{NI} + d - \phi_{NI} \right]^2 \]
Least Squares Interpolation

leastSquaresVolPointInterpolation class

\[
\begin{bmatrix}
\sum_{i=1}^{N} x_{N_i}^2 & \sum_{i=1}^{N} x_{N_i} y_{N_i} & \sum_{i=1}^{N} x_{N_i} z_{N_i} & \sum_{i=1}^{N} x_{N_i} \\
\sum_{i=1}^{N} x_{N_i} y_{N_i} & \sum_{i=1}^{N} y_{N_i}^2 & \sum_{i=1}^{N} y_{N_i} z_{N_i} & \sum_{i=1}^{N} y_{N_i} \\
\sum_{i=1}^{N} x_{N_i} z_{N_i} & \sum_{i=1}^{N} y_{N_i} z_{N_i} & \sum_{i=1}^{N} z_{N_i}^2 & \sum_{i=1}^{N} z_{N_i} \\
\sum_{i=1}^{N} x_{N_i} & \sum_{i=1}^{N} y_{N_i} & \sum_{i=1}^{N} z_{N_i} & N
\end{bmatrix}
\begin{bmatrix}
a \\
b \\
c \\
d
\end{bmatrix} =
\begin{bmatrix}
\sum_{i=1}^{N} \phi_{N_i} x_{N_i} \\
\sum_{i=1}^{N} \phi_{N_i} y_{N_i} \\
\sum_{i=1}^{N} \phi_{N_i} z_{N_i} \\
\sum_{i=1}^{N} \phi_{N_i}
\end{bmatrix}
\]

Solve this linear system using Gaussian elimination implemented in simpleMatrix class
Special treatment of boundaries

Coupled boundaries uses neighbour cell centre values, and empty patch values are included so A is not singular
Comparison of Inverse Distance & Least Squares Methods
Comparison of Inverse Distance & Least Squares Methods

Inverse Distance

Least Squares
Comparison of Inverse-Distance & Least Squares Methods

Inverse Distance

Least Squares
Comparison of Inverse-Distance & Least Squares Methods

Rotation: 0

Inverse Distance

Least Squares
Summary & Conclusions

An updated Lagrangian finite volume structural solver suitable for large rotations has been developed.

Boundary non-orthogonal correction is imperative.

Standard inverse distance interpolation shows poor performance on non-orthogonal grids.

Therefore, a least squares approach has been developed.
Development of a Finite Volume Based Structural Solver for Large Rotation of Non-Orthogonal Meshes

Philip Cardiff

Thanks

philip.cardiff@ucd.ie
philip.cardiff@gmail.com

Alojz Ivanković
Aleksandar Karač
Željko Tuković