ALE Mesh Updating using Mesh Optimisation

Alan Kelly[1], Lukasz Kaczmarczyk[2], Ross Mackenzie[3], and Chris Pearce[4]

[1] University of Glasgow a.kelly.2@research.gla.ac.uk
[2] University of Glasgow Lukasz.Kaczmarczyk@glasgow.ac.uk
[3] University of Glasgow r.mackenzie.2@research.gla.ac.uk
[4] University of Glasgow Chris.Pearce@glasgow.ac.uk

1 Introduction

Accurate numerical simulation of problems in fluid mechanics requires using a method whereby the mesh can track free surfaces or fluid-solid or fluid-fluid interfaces whilst coping with large distortions of the continuum. Such methods are referred to as Arbitrary Lagrangian Eulerian (ALE) methods and may be furthered studied in [1]. In order for the mesh to adapt to the deforming domain, a mesh update procedure must be performed at each time step of a calculation. In this paper we focus on a mesh update technique which aims to maintain or even improve the quality of a deforming mesh whilst accurately tracking the domain boundary. We focus on a problem from microfluid dynamics in order to illustrate our proposed method.

2 Mesh Optimisation

2.1 Quality Measure

Finding a suitable quality measure that provides an accurate estimate of an element's effects in terms of discretisation/interpolation error and stiffness matrix condition is challenging and is a very active area of research in itself. There are many measures in existence and these may be further studied in [2] and [3]. A triangular element has many properties which determine its effect on the accuracy of a FEM simulation, however the interior angles of the element have been shown to be of greatest importance. Of the many measures in existence, very few directly eliminate deleterious interior angles from meshes. Instead, they aim to modify some other quantity which is indirectly linked to an element's interior angles. In theory, a measure which determines new nodal positions based on an element's interior angles should produce meshes which are far more suited for use in Finite Element (FE) simulations. Several measures already exist which operate directly on an element's interior angles, such as using the sine of the interior angles of element as a quality measure. In theory, this measure should be effective. However, Figure 1 demonstrates why this is not so. A typical objective function would sum the sines (or the square of the sines) of every angle in the solution space, that is the set of angles which are effected by the movement of a node. When the selected node is moved in the triangular mesh, Figure 1, 18 angles are affected. If one angle is made worse or even negative, the objective function may not reflect this as the contributions of the other angles in the solution space overpower it. [4] proposed a solution involving non-smooth optimisation referred to as the minimum sine measure, which seeks to maximise the minimum sine of the angles in a mesh. This involves selecting the worst angle in the solution space and moving the free node in the direction of greatest improvement until it is no longer the worst angle. This algorithm is included in Mesquite [5] and Stellar [6], both showing it to be effective. However, a smooth objective function in theory should yield much better results as the full range of numerical optimisation tools are available.

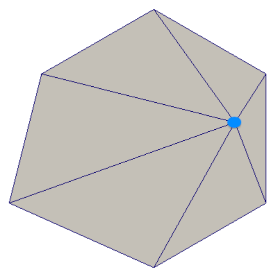

Fig. 1. When the free node is moved, 18 angles are affected.

In order to address the deficiencies of existing quality measures, we propose a new quality measure which involves using a log-barrier as an objective function and expresses every angle as a function of the worst angle in the solution space. This takes the following form:

$$q = \frac{sin^2(\theta)}{2(1-\gamma)} - log(sin(\theta) - \gamma) \tag{1}$$

where θ is the angle and γ is the worst angle in the solution space. Each angle in the solution space is weighted logarithmically according to its sine; the smaller the sine, the greater the weight. This is shown graphically in Figure 2. The use of a log-barrier is explained in detail in [7].

2.2 Boundary Nodes

Elements containing boundary nodes introduce additional complexity to the optimisation process. Nodes which lie on the boundary of the domain can only be moved in a manner which respects the geometry and volume of the domain. The geometry of the domain is determined by the physical processes being modelled and thus must be respected; the volume must also be conserved. Nodes which lie on surfaces which may be mathematically defined are

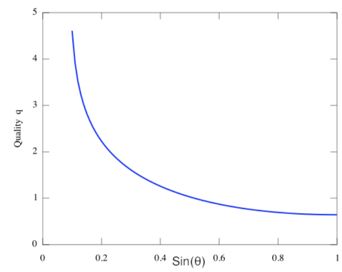

Fig. 2. Log Barrier

straightforward to move. However, nodes which lie on more complex surfaces are much more difficult to move in a manner which respects both volume and geometry. The authors have developed a method for moving such nodes which is clearly explained and demonstrated in three-dimensions in [7]. A two-dimensional version of this algorithm is currently under development.

3 Applications

Computational mechanics is often used to study problems with evolving geometries, for example problems involving fracture, microfluids, surface tension and crystal growth. Such problems are usually solved by applying an ALE formulation which requires the use of mesh updating procedures. These update procedures may include remeshing, modifying mesh topology and mesh smoothing. Remeshing and topological modifications introduce large numerical errors at the beginning of each time step making convergence difficult to achieve. For this reason, we wish to limit remeshing and topological modifications during the simulation process. Therefore we propose using mesh optimisation techniques to determine the optimal nodal positions at every time step by coupling the mesh optimisation equations with the physical equations. Coupling is the process of solving the physical equations and the mesh optimisation equations simultaneously. Coupling the mesh optimisation equations with the physical equations results in a more efficient technique which gives the user control over the numerical errors and allows for quadratic convergence when an implicit method, such as a Newton solver, is used. We believe that such techniques will allow the use of larger time steps and give us stable control over numerical errors. Adaptive time stepping is used to determine the largest possible time step to ensure stability of the simulation. Too small a time step and the simulation will be too slow; too large a time step and the solver will fail to converge. Therefore, any measure which can safely increase the time step is worth implementing.

To demonstrate the application of mesh optimisation to such problems, we shall apply the quality measure introduced in Section 2.1 to a two dimensional axisymmetric model of a fluid droplet which we have developed that can accurately predict the response of microfluid droplets subjected to gravitational forces and surface acoustic waves. This droplet undergoes massive deformations, Figure 3, and thus the mesh must adapt to the deforming domain.

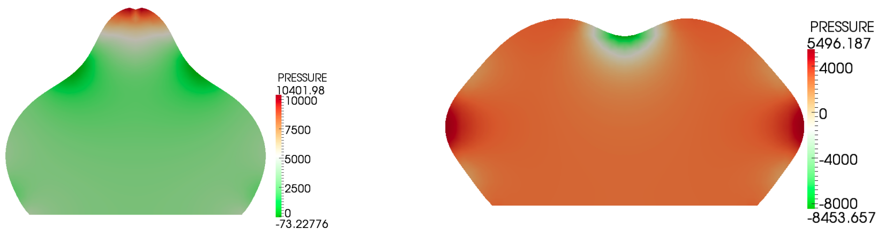

Fig. 3. Deformed microfluid droplet with the pressure distribution plotted

3.1 Mesh Optimisation

The original implementation of this simulation involved the use of Laplacian smoothing. The idea behind Laplacian smoothing is that the nodes of high quality mesh are separated by equal distances. However, the inverse is not true - good mesh need not consist of equidistant nodes. As a result, Laplacian smoothing can induce excessive and unnecessary nodal displacements in initially good mesh. Every nodal displacement generates non-linearities which negatively effect the rate of convergence and the efficiency of the calculation. Another reason why we require an alternative to Laplacian smoothing is that although it is somewhat effective in 2D, it is not at all effective in 3D. When the 2D model performs satisfactorily, the next step will be to implement it in 3D; thus techniques which will translate to three dimensions are required.

4 Results and Problems

The mesh optimisation algorithm was first verified in a decoupled form using the meshes in Figure 3. In both cases much better meshes were obtained. The nodes on the curved surface were assumed to be fixed. Once verified, the optimisation equations were coupled with the physical equations and the results compared with those obtained using Laplacian smoothing. This simulation was run until the droplet reached its equilibrium position. The mesh update algorithms are judged by four criteria: the number of times remeshing was required, the percentage volume loss, the number of time steps taken to reach the equilibrium position and the range of interior angles in the mesh during the entire simulation. The results shown in Table 1 show that our mesh optimisation algorithm had the desired effect of reducing the need to remesh. The loss of minute quantities of volume has been a problem throughout the development process and in this case the application of our mesh optimisation techniques has reduced the volume loss by almost three-quarters compared to Laplacian smoothing. The simulation using Laplacian smoothing reached the equilibrium position in fewer time steps than the one using mesh optimisation. In fact, Laplacian smoothing reached the equilibrium position in

Table 1. Comparison of Laplacian Smoothing and Mesh Optimisation

Mesh Update Algorithm	Times Remeshed	Volume Loss	Number of Time Steps	Range of Angles
Laplacian Smoothing	533	-0.139%	3200	8.07°-163.06°
Mesh Optimisation	277	-0.036%	5100	23.5°-115.03°

approximately half the time taken by mesh optimisation as each of mesh optimisation's time steps took longer to solve. This can be partly explained by the time taken to calculate the mesh quality contributions to the stiffness matrix and force vector. However, this is not a satisfactory explanation and thus warrants further investigation. A much higher quality mesh was obtained by mesh optimisation throughout the entire simulation process as can be seen in Table 1. This shows that our mesh optimisation scheme is much more effective than Laplacian smoothing at ensuring that the mesh throughout the entire simulation is suitable for the calculations being performed on it.

5 Future Work

This project is very much a work in progress. The next step is to determine the reasons for the simulation run with mesh optimisation taking significantly longer than with Laplacian smoothing. We also wish to add the smoothing of surface nodes to our model. We believe that this will greatly reduce the need to remesh, thus giving us more control over the numerical errors. Once this model performs satisfactorily in 2D, the next step will be to implement it in 3D.

References

1. J Donea, Antonio Huerta, J Ponthot, and A Rodr. *Encyclopedia of Computational Mechanics.* John Wiley & Sons, Ltd, Chichester, UK, November 2004.
2. Jonathan Richard Shewchuk. What is a Good Linear Element ? Interpolation , Conditioning , and Quality Measures. In *Proceedings of the 11th International Meshing Roundtable*, 2002.
3. Patrick M. Knupp. Algebraic Mesh Quality Metrics. *SIAM Journal on Scientific Computing*, 23(1):193, 2001.
4. Lori a. Freitag, Mark Jones, and Paul Plassmann. An Efficient Parallel Algorithm for Mesh Smoothing. In *Proceedings of the 4th International Meshing Roundtable*, 1995.
5. Patrick Knupp, Lori Freitag-Diachin, and Jason Kraftcheck. Mesh Quality Improvement Toolkit User's Guide. 2008.
6. B Klingner. *Tetrahedral mesh Improvement.* PhD thesis, University of California at Berkeley, November 2008.
7. Alan Kelly, Lukasz Kaczmarczyk, and Chris Pearce. Mesh improvement methodology for 3d volumes with non-planar surfaces, 2013. http://arxiv.org/abs/1307.3404.