The University Of Reading

Generation of Arbitrary Lagrangian-Eulerian (ALE) velocities, based on monitor functions, for the solution of compressible fluid equations.

B. V. Wells, M. J. Baines and P. Glaister.

27th April 2004

Numerical Analysis Report 8/04

Presented at the ICFD conference on numerical methods for fluid dynamics
Oxford, UK. 29 March - 1 April 2004.

DEPARTMENT OF MATHEMATICS

Abstract

A moving mesh method is outlined based on the use of monitor functions. The method is developed from a weak conservation principle. From this principle a conservation law for the mesh position is derived. Using the Helmholtz decomposition theorem, this conservation law can be converted into an elliptic equation for a mesh velocity potential.

The moving mesh method is discretised using standard finite elements. Once the mesh velocities are obtained an Arbitrary Lagrangian Eulerian (ALE) [3] fluid solver is used to update the solution on the adaptive mesh.

Results are shown for the compressible Euler equations of gas dynamics in one and two spatial dimensions. Two monitor functions are used, the fluid density (which corresponds to a Lagrangian description), and a function which includes the density gradient. A variety of test problems are considered.

1 INTRODUCTION

In this paper an adaptive method is presented for the solution of multi-dimensional hyperbolic conservation laws of the form

$$\mathrm{U}_t + \nabla \cdot F(\mathrm{U}) = 0 \quad \text{in } \Omega(t) \times [0, T]. \tag{1}$$

Here $\mathrm{U} \equiv \mathrm{U}(\mathrm{x}, t)$ is some m-vector of conserved variables and x and t are spatial and temporal variables respectively. The system (1) is solved in some spatial region $\Omega(t)$ having a boundary $\partial\Omega(t)$ which may or may not be moving in time.

2 FORMULATION OF THE MOVING MESH METHOD

Let $M(\mathrm{x}, t) > 0$ be a user-defined monitor function which reflects the characteristics of the solution to (1). The mesh positions are chosen to satisfy

$$\int_{\Omega(t)} M \, \mathrm{d}\Omega = \text{constant in time}, \tag{2}$$

for all $\Omega(t)$. In practice M will depend on the solution of the PDE (1) and its partial derivatives. Differentiating (2) with respect to time gives

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega(t)} M \, \mathrm{d}\Omega = 0, \tag{3}$$

which is a Lagrangian description of a pseudo-fluid which has a pseudo-density function M and moves with a pseudo-velocity $\dot{\mathrm{x}}$, say. Using the Reynold's Transport Theorem on (3) gives

$$\int_{\Omega(t)} \nabla \cdot (M \, \dot{\mathrm{x}}) \, \mathrm{d}\Omega = - \int_{\Omega(t)} M_t \, \mathrm{d}\Omega, \tag{4}$$

which is a conservation law for the pseudo-fluid. The velocity $\dot{\mathrm{x}}$ is not uniquely determined by equation (4). Therefore an additional condition on the velocity field is required. Following Cao, Huang and Russell in [2], the Helmholtz decomposition theorem is used to prescribe the curl condition

$$curl \, \dot{\mathrm{x}} = curl \, \mathrm{q}, \tag{5}$$

where q is some given vector field. The curl condition shows that there exists a potential function ψ such that

$$\dot{\mathrm{x}} = \mathrm{q} + \nabla\psi. \tag{6}$$

Using (6) in equation (4) results in an elliptic equation for the velocity potential ψ of the pseudo-fluid.

$$\int_{\Omega(t)} \nabla \cdot (M\,\nabla\psi)\,\mathrm{d}\Omega = -\int_{\Omega(t)} M_t\,\mathrm{d}\Omega - \int_{\Omega(t)} \nabla \cdot (M\,\mathrm{q})\,\mathrm{d}\Omega \tag{7}$$

A boundary condition involving either ψ or $\frac{\partial\psi}{\partial n}$ on the boundary of the domain $\partial\Omega$ is also required. One such condition which may be prescribed is $\dot{\mathrm{x}} \cdot \mathrm{n} = 0$ or $\frac{\partial\psi}{\partial n} = -\mathrm{q} \cdot \mathrm{n}$, that is none of the pseudo-fluid should leave the domain. Since $M > 0$ and M_t is prescribed in terms of the solution of the underlying PDE (1), equation (7) has a unique solution ψ from which $\dot{\mathrm{x}}$ can be obtained using (6). The velocity of the pseudo-fluid $\dot{\mathrm{x}}$ is induced by the choice of the monitor function M and the velocity field q. It is this velocity $\dot{\mathrm{x}}$ that will become the ALE velocity.

3 WEAK FORMS AND A FINITE ELEMENT METHOD

The velocity potential equation (7) obtained in the previous section now needs to be solved numerically for ψ and hence the velocity of the pseudo-fluid $\dot{\mathrm{x}}$. This velocity will then be used as the ALE velocity in an ALE fluid solver. The velocity potential equation will be solved using finite elements, as this method is very flexible in multidimensions [1]. A weak form of equation (7) will be needed so that we can apply a FE discretisation. A distributed generalisation of equation (2) is

$$\int_{\Omega(t)} w\,M\,\mathrm{d}\Omega = \text{constant in time}, \tag{8}$$

where w is a test function, which is once differentiable. Upon differentiating (8) with respect to time and using equation (6), we have

$$\int_{\Omega(t)} w\nabla \cdot (M\,\nabla\psi)\,\mathrm{d}\Omega = -\int_{\Omega(t)} w\,M_t\,\mathrm{d}\Omega - \int_{\Omega(t)} w\nabla \cdot (M\,\mathrm{v})\,\mathrm{d}\Omega. \tag{9}$$

The test function w is chosen to be moving with the velocity $\dot{\mathrm{x}}$ and hence $\frac{Dw}{Dt} = \frac{\partial w}{\partial t} + \dot{\mathrm{x}} \cdot \nabla w = 0$, which has been used in the derivation of (9). Now that a weak form of the velocity potential equation has been derived, we can apply standard finite elements.

We denote the finite element approximations to U, $\dot{\mathrm{x}}$, v and ψ by $\tilde{U}$, $\dot{\mathrm{X}}$, V and Ψ respectively. The domain $\Omega(t)$ is partitioned into nonoverlapping computational cells and a patch of such cells surrounding the i^{th} node will be denoted by $\Omega_i(t)$. The test function w becomes one of the finite element basis functions w_i which form a partition of unity. All the finite element approximations are expanded in terms of these basis functions. In all the work presented we will take the basis functions to be piecewise linear functions, so hat functions in 1D and pyramid functions defined on triangular elements in 2D. Therefore the velocity potential equation (9) becomes

$$\int_{\Omega_i(t)} w_i \nabla \cdot (M\, \nabla\Psi)\, \mathrm{d}\Omega = -\int_{\Omega_i(t)} w_i\, M_t\, \mathrm{d}\Omega - \int_{\Omega_i(t)} w_i \nabla \cdot (M\, \mathrm{V})\, \mathrm{d}\Omega. \quad (10)$$

The velocity potential equation (10) leads to a weighted stiffness matrix system for Ψ. The velocity is then recovered from the velocity potential using the weak form

$$\int_{\Omega_i(t)} w_i\, \dot{\mathrm{X}}\, \mathrm{d}\Omega = \int_{\Omega_i(t)} w_i\, \nabla\Psi\, \mathrm{d}\Omega, \quad (11)$$

which is equivalent to

$$\min_{\dot{\mathrm{X}}_i} ||\dot{\mathrm{X}} - \nabla\Psi||^2_{L_2}. \quad (12)$$

Equation (12) leads a set of mass matrix systems, one for each component of the velocity field $\dot{\mathrm{X}}$. Once the velocity has been found we can use it in an ALE fluid solver and time-step the mesh.

4 NUMERICAL RESULTS

In this section, numerical results are presented for the solution of the two-dimensional compressible Euler equations. The Euler equations, along with the ideal gas equation of state, are solved on a moving adaptive mesh generated by the method outlined in the previous sections. In all the results shown the ratio of specific heats was taken to be $\gamma = 1.4$. Once a mesh velocity has been generated, by solving (10) and calculating the velocity through (12), the Euler equations need to be solved on a moving mesh. This is done by solving the ALE form of the Euler equations, which are the equations transformed into a moving frame of reference. In one spatial dimension this is done with the ALE form of the Roe Riemann solver [4, 5] and in two dimensions by

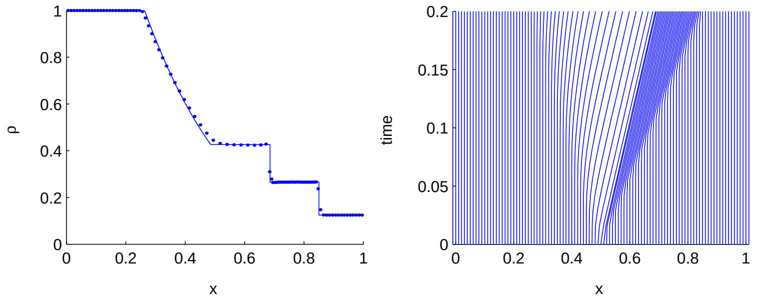

Figure 1: Lagrangian Solution to the 1D Sod shock tube problem at $t = 0.2$ using a Roe Riemann-solver with a superbee limiter on a moving mesh obtained from $M = \rho$. $(CFL = 0.5, N = 100)$. Trajectories of the mesh points.

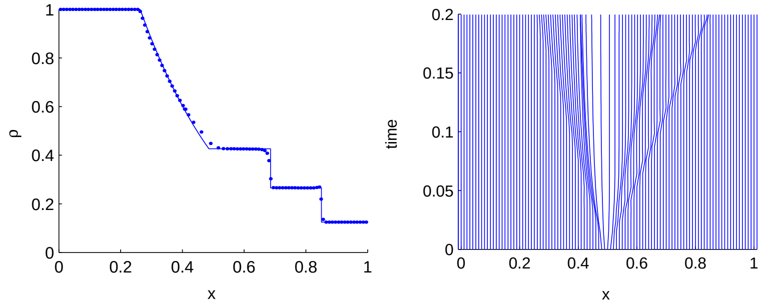

Figure 2: ALE Solution to the 1D Sod shock tube problem at $t = 0.2$ using a Roe Riemann-solver with a superbee limiter on a moving mesh obtained from $M = 1 + \alpha|\rho_x|$, $\alpha = \frac{\beta}{\max\limits_x |\rho_x|}$. $(CFL = 0.5, N = 100, \beta = 2)$. Trajectories of the mesh points.

the HLLC Riemann solver [7]. It should also be noted that in all the results shown we have taken the velocity field q to be zero.

Two test cases are considered. In 1D we solved the Sod shock tube problem [6]. The problem consists of two regions of gas initially separated by a membrane at $x = \frac{1}{2}$. The gas to the left of the membrane is more dense and at a higher pressure compared to the one on the right. Also, the gases are initially at rest. The membrane is then removed and the problem is to find the subsequent motion of the gases.

This problem was solved using two different monitor functions. The first used was $M = \rho$, where ρ is the density of the gas. This monitor function leads to Lagrangian mesh movement. Results for this monitor function are shown in figure 1. We also solved this problem using the monitor function

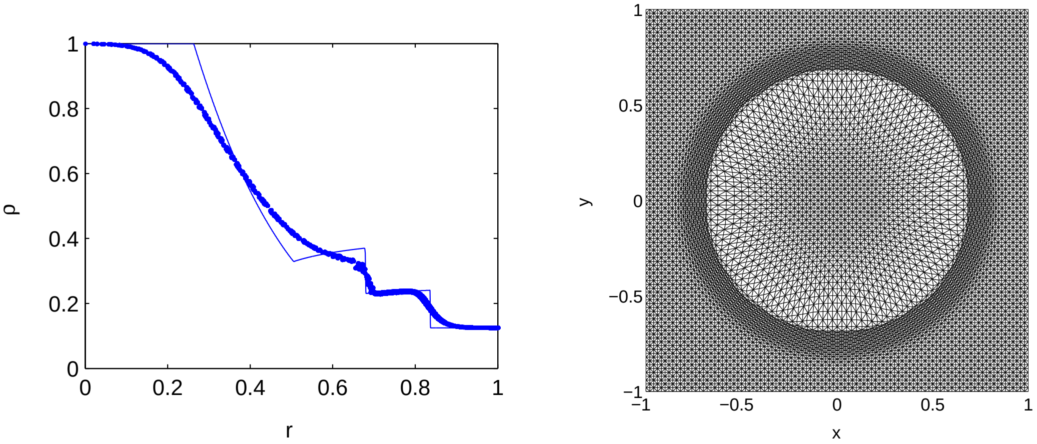

Figure 3: Density solution to the diverging cylindrical shock tube problem at $t = 0.2$ using a first order in space HLLC Riemann-solver on a moving mesh obtained from $M = \rho$. Mesh with 10201 nodes and 20000 triangles obtained at $t = 0.2$.

$M = 1 + \alpha|\rho_x|$, with a suitable scaling α. This monitor function was chosen so as to move mesh points into regions of the flow where there are large variations in the fluid density. Results for this problem are shown in figure 2. The trajectories clearly show how the mesh follows the velocity of the fluid in figure 1 and responds to the characteristics in figure 2. The results obtained from the moving mesh algorithm were compared to the exact solution, which was computed with an exact Riemann solver.

In 2D we solved a cylindrical shock problem. The problem consists of two regions of gas separated by a membrane at $x^2 + y^2 = \left(\frac{1}{2}\right)^2$. The gas at the centre of the region has a higher density and is at a higher pressure compared to the one outside the membrane.

This problem was also solved using two different monitor functions. The first monitor function was again chosen to used was $M = \rho$. Results for this monitor function can be seen in figure 3. The second monitor function used was $M = 1 + \alpha|\nabla\rho|^2$, where α was suitably chosen. Results for this monitor function can be seen in figure 4. The problem was solved numerically in 2D for each of these monitor functions and compared with a 1D radial computation of the problem computed on a very fine mesh. Although the density profiles are not strongly affected the mesh is clearly moving in a rational way. It is expected that considerable improvement will occur when a second order solver is implemented.

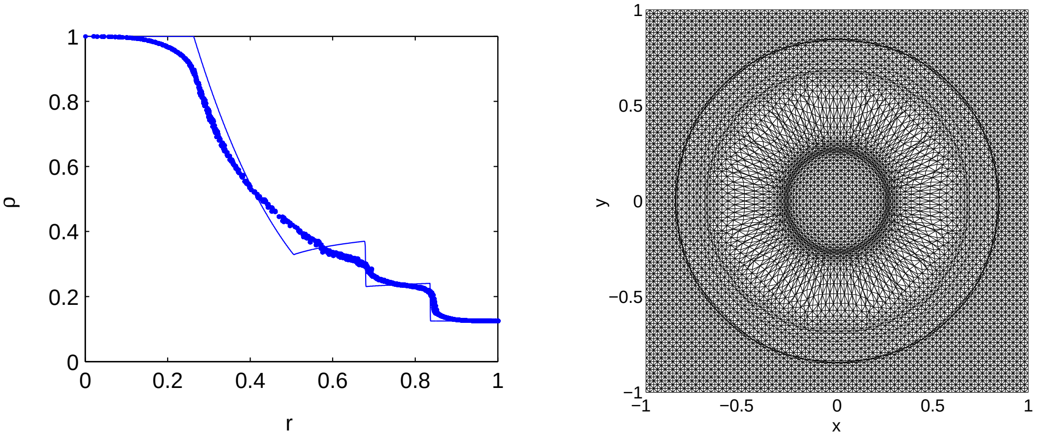

Figure 4: Density solution to the diverging cylindrical shock problem at $t = 0.2$ using a first order in space HLLC Riemann-solver on a moving mesh obtained from $M = 1 + \alpha|\nabla\rho|^2$. Mesh with 10201 nodes and 20000 triangles obtained at $t = 0.2$.

5 CONCLUSIONS

A method for generating mesh velocities using monitor functions has been presented which can then be used in ALE fluid solvers. We have used the method outlined in sections 2 and 3 to produce an adaptive mesh for the solution of the Euler equations of gas dynamics in one and two spatial dimensions. Two test problems have been solved using different monitor functions leading to different types of mesh movement.

In future work we aim to use improved initial meshes for the problem, instead of equi-spaced meshes. We also need to increase the order of the HLLC Riemann solver in 2D in order to better resolve the density profile, which will in turn sharpen the adaptivity. Other monitor functions will be tried and work will also be done on investigating the influence of the rotational vector field q in (6) and how to choose it for a given problem.

References

[1] K.W. Blake: *Moving Mesh Methods for Non-Linear Parabolic Partial Differential Equations.* Ph.D thesis, Department of Mathematics, University of Reading, 2001

[2] W. Cao, W. Huang and R.D. Russell: *A Moving Mesh Method Based On The Geometric Conservation Law.* Siam Journal On Scientific Computing. Vol. 24, No. 1, 118-142 (2002)

[3] C.W. Hirt, A.A. Amsden and J. L. Cook: *An Arbitrary Lagrangian-Eulerian Computing Method For All Flow Speeds.* Journal of Computational Physics 14, 227 (1974)

[4] P.L. Roe: *Approximate Riemann Solvers, Parameter Vectors, And Difference Schemes.* Journal of Computational Physics 135, 250-258 (1981)

[5] P.L. Roe: *Some Contributions Of The Modelling Of Discontinuous Flows.* Lecture Notes In Applied Mathematics, 163-193 (1985)

[6] G.A. Sod: *A Survey Of Several Finite Difference Methods For Systems Of Nonlinear Hyperbolic Conservation Laws.* Journal of Computational Physics 27, 1-31 (1978)

[7] E.F. Toro, M. Spruce and W. Speares: *Restoration Of The Contact Surface In The HLL-Riemann Solver.* Shock Waves 4, 25-34 (1994)