INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2002; **40**:3–19 (DOI: 10.1002/fld.294)

Moving meshes, conservation laws and least squares equidistribution

M. J. Baines*

Department of Mathematics, The University of Reading, Reading RG6 6AX, UK

SUMMARY

In this paper a least squares measure of a residual is minimized to move an unstructured triangular mesh into an optimal position, both for the solution of steady systems of conservation laws and for functional approximation. The result minimizes a least squares measure of an equidistribution norm, which is a norm measuring the uniformity of a fluctuation monitor. The minimization is carried out using a steepest descent approach. Shocks are treated using a mesh with degenerate triangles. Results are shown for a steady-scalar advection problem and two flows governed by the Euler equations of gasdynamics. Copyright © 2002 John Wiley & Sons, Ltd.

KEY WORDS: moving meshes; conservation laws; least squares equidistribution

1. INTRODUCTION

Meshes are a significant part of CFD and heat transfer calculations. One challenge is to extract the optimal benefit from the mesh, whether in terms of the accuracy of the solution or a relevant physical measurement. Mesh movement and mesh subdivision are the prime candidates: here we discuss mesh movement.

The governing equations of CFD and heat transfer arise from the balance laws of mass, momentum and energy written in a suitable reference frame. In order to write the fluid variables as functions of the spatial variables it is necessary to specify the frame, e.g. Eulerian (fixed mesh), Lagrangian (mesh moving with the fluid velocity) or some other frame, perhaps related to mesh quality, error minimization or equidistribution. Realistically, there is only one choice of viewing frame for all the dependent variables in a problem and this is also the case when designing mesh movement.

The main difficulties associated with mesh movement are the complexity of solving the non-linear mesh equations and the problem of mesh tangling. A useful approach is to generate the mesh (and even the solution) locally node by node and carry out sweeps through the mesh, greatly reducing the complexity of individual steps and allowing much better control of mesh tangling.

*Correspondence to: M. J. Baines, Department of Mathematics, The University of Reading, P.O. Box 220, Reading RG6 6AX, U.K.

Received May 2001
Revised November 2001

Many mesh movement algorithms depend on optimization, using functionals measuring for example mesh quality or error measures or some other desirable feature [1, 2]. For problems derived from a variational principle both the mesh and the approximate solution can be determined from the same functional by optimizing in an appropriate finite-dimensional space [3–5].

For problems which cannot be derived from a variational principle simultaneous generation of the mesh and solution can be achieved by using a least squares functional in lieu of a variational principle, albeit with some extra complications. Meshes and solutions for simple flow problems have been generated in this way, using either global or local methods of solution [6–8].

For more realistic problems in CFD and heat transfer, separate principles for the mesh and the solution are usually applied, allowing the full power of standard algorithms to be used for the solution [9]. For example, fluctuation distribution finite volume schemes may be used in conjunction with mesh movement optimization and this is the programme followed here Reference [10]. Since the mesh principles naturally involve the solution of the governing equations, the two principles do not decouple.

In this paper we shall motivate and develop an approach to the generation of unstructured triangular meshes for conservation laws based on minimizing the least squares error of fluctuations. The same principle can also be applied to the problem of function equidistribution using a vector-valued monitor function and this is discussed in some detail. A descent method is used to carry out the minimizations, generating a simple adaptive mesh movement rule for function equidistribution and giving insights into mesh movement for fluctuation equidistribution. The approach may be combined with a flux distribution finite volume scheme to obtain the solution of steady-state problems on optimal meshes within an iterative algorithm. Whilst the use of modern finite volume schemes is preferable to a least squares approach when determining the solution, there is a role for mesh and solution least squares minimizations in a shock fitting context when moving the mesh to adjust the position of shocks using degenerate triangles [11].

The layout of the paper is as follows. After sections on conservation laws and the status of fluctuation splitting when the mesh is allowed to move, we describe the mesh movement mechanism that we use. This is based on l_2 minimization and is motivated by its equivalence with the minimization of an equidistribution norm in a least squares sense [12]. A similar argument is applicable to the problem of function approximation in several dimensions using a vector-valued equidistributed quantity. Descent methods are described in both cases. Results are shown for a scalar problem and for a problem governed by the Euler equations of gasdynamics, both using least squares for the mesh in conjunction with a fluctuation splitting scheme for the solution, as well as for an Euler equations case using least squares on degenerate triangles.

2. SYSTEMS OF CONSERVATION LAWS

A system of conservation laws (e.g. the Euler equations) can be written as

$$\operatorname{div} \underline{\mathbf{f}}(\mathbf{u}(\underline{x})) = 0 \tag{1}$$

where $\mathbf{u}(\underline{x})$ is the vector of conserved variables and $\underline{\mathbf{f}}(\mathbf{u}(\underline{x}))$ is the flux function. The flux function is a known function of $\mathbf{u}$ while $\mathbf{u}(\underline{x})$ is to be determined.

Integral forms of the conservation law are

$$\oint_{\partial\Omega} \underline{\mathbf{f}}(\mathbf{u}(\underline{x})).\hat{\underline{n}}\,\mathrm{d}\sigma = \int_{\Omega} \operatorname{div} \underline{\mathbf{f}}(\mathbf{u}(\underline{x}))\,\mathrm{d}\Omega = 0 \tag{2}$$

where $\mathrm{d}\sigma$ is an element of the boundary $\partial\Omega$ of Ω and $\hat{\underline{n}}$ is the inward unit normal to $\partial\Omega$.

3. FLUCTUATIONS

Let the plane polygonal domain Ω be divided into triangles T_k and let $\mathbf{U}_j$ be an approximate nodal solution value at a vertex j, which depends on the set of nodal positions $\{\underline{X}\}$ in the triangulation. Let the function $\underline{\mathbf{f}}(\mathbf{u}(\underline{x}))$ be approximated by a *continuous* piecewise linear function $\underline{\mathbf{F}}$ with vertex values $\underline{\mathbf{F}}_j = \underline{\mathbf{f}}(\mathbf{U}_j)$. The values of $\mathbf{U}_j(\{\underline{X}\})$ are assumed to be known or computable at each vertex.

We define a *fluctuation* in triangle T_k to be

$$\mathbf{\Phi}_k = \oint_{\partial T_k} \underline{\mathbf{F}}.\hat{\underline{n}}\,\mathrm{d}\sigma = -\int_{T_k} \operatorname{div} \underline{\mathbf{F}}\,\mathrm{d}S \tag{3}$$

(cf. (2)), where ∂T_k is the perimeter of T_k and $\mathrm{d}S$ is an element of area. Since $\underline{\mathbf{F}}$ is piecewise linear and $\operatorname{div}\underline{\mathbf{F}}$ is piecewise constant we may write the fluctuation as

$$\mathbf{\Phi}_k = \left(\sum_{v=1}^{3} \frac{1}{2}(\underline{\mathbf{F}}(\mathbf{U}(\underline{X}_{v1})) + \underline{\mathbf{F}}(\mathbf{U}(\underline{X}_{v2})))\underline{N}_v \right)_k = -S_k(\operatorname{div}\underline{\mathbf{F}})_k \tag{4}$$

where $v1, v2$ are the indices of the vertices in the triangle k other than v, and the Δ operator indicates the change in the argument between the ends of the side *opposite* node v, taken anticlockwise. The vector

$$\underline{N}_v = (-(\Delta Y)_v, (\Delta X)_v)^{\mathrm{t}} \tag{5}$$

is the normal to the side opposite the vertex, measured inward, having the length of that side (see Figure 1), with the property $\sum \underline{N}_v = 0$, and S_k is the area of triangle T_k.

We thus have from the first of (4)

$$\mathbf{\Phi}_k = -\frac{1}{2}\sum_{v=1}^{3} \underline{\mathbf{F}}(\mathbf{U}(\underline{X}_v)) \begin{pmatrix} -\Delta Y \\ \Delta X \end{pmatrix}_v = \frac{1}{2}\sum_{v=1}^{3} (\Delta\underline{\mathbf{F}}(\mathbf{U}(\underline{X})))_v \begin{pmatrix} -Y \\ X \end{pmatrix}_v \tag{6}$$

using (5) and summation by parts.

4. FLUCTUATION SPLITTING

In fluctuation splitting schemes [13] for steady problems the fluctuations are regarded as error measures (similar to residuals) and an iterative procedure is set up which adds a multiple of

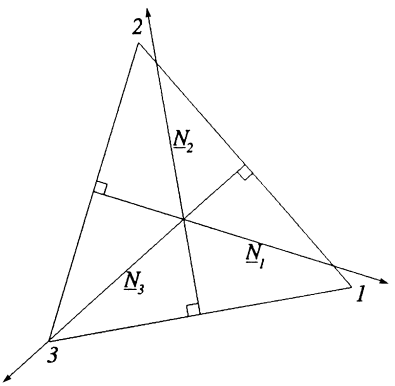

Figure 1. A general triangular cell.

the fluctuation to the vertices of the triangle with prescribed weights, with a view to driving them down to zero. In the well-known multidimensional upwind schemes [14, 15] the weights are chosen so that the contribution to upwind vertices is zero and the schemes are conservative, positive and linearity preserving. In particular, conservation is assured if the weights in each triangle sum to unity.

The number of fluctuations is equal to the number of triangles in the mesh but the number of unknowns is a multiple of the number of nodes, which is different in general. When the number of fluctuations exceeds the number of unknowns it is impossible to drive all the fluctuations to zero. For fluctuation distribution schemes convergence of the nodal updates does not then imply that the fluctuations vanish in each cell [8].

If however, we allow the co-ordinates of the vertices to become additional unknowns of the problem, the solution is improved. For scalar problems the number of unknowns then typically exceeds the number of fluctuations and there are infinitely many configurations which make the fluctuations zero. A unique solution is achieved in a two-dimensional scalar problem on a simple mesh by including just one co-ordinate per node in the list of unknowns so that the number of unknowns is equal to the number of equations [8]. The fluctuations may then be driven down to zero by a fluctuation splitting scheme and the result is an approximate method of characteristics (cf. Reference [8]). For a system of two equations in two dimensions on a simple mesh, with the nodes allowed to move freely, the number of unknowns is equal to the number of equations and this has been studied in Reference [6]. For systems such as the Euler equations the number of fluctuations remains less than the number of unknowns, but the inclusion of nodal variables significantly reduces the dimension of the null space.

There are many mechanisms available for moving the nodes, including mechanical analogues and optimization. Here we implement an approximate equidistribution principle based on least squares minimization, using either the fluctuations themselves or a monitor function.

5. LEAST SQUARES EQUIDISTRIBUTION

5.1. Fluctuations

Consider the identity

$$N\sum_{k=1}^{N}\boldsymbol{\Phi}_k^{\mathrm{t}}Q\boldsymbol{\Phi}_k \equiv \left(\sum_{k=1}^{N}\boldsymbol{\Phi}_k\right)^{\mathrm{t}} Q\left(\sum_{k=1}^{N}\boldsymbol{\Phi}_k\right) + \sum_{k=1}^{N}\sum_{l=1}^{N}(\boldsymbol{\Phi}_k-\boldsymbol{\Phi}_l)^{\mathrm{t}}\,Q(\boldsymbol{\Phi}_i-\boldsymbol{\Phi}_l) \tag{7}$$

where Q is any matrix weight. The sum in the first term on the right-hand side of (7) is

$$\sum_{k=1}^{N}\boldsymbol{\Phi}_k = \sum_{k=1}^{N}\oint_{\partial T_k}\underline{\mathbf{F}}.\hat{\underline{n}}\,\mathrm{d}\sigma = \oint_{\partial\Omega}\underline{\mathbf{F}}.\hat{\underline{n}}\,\mathrm{d}\sigma \tag{8}$$

using cancellation at internal edges. This is a small quantity (cf. (2)) but more importantly is independent of interior values of $\underline{\mathbf{F}}$ and interior grid locations $\underline{X}$. Hence from (7) the weighted least squares norm of the fluctuations on the left-hand side of (7) and the weighted norm of the fluctuation *differences* on the right-hand side of (7) are minimized simultaneously over *interior* parameters. This is Least squares Equidistribution [12].

If complete equalization of the fluctuations is possible, then the norm of the fluctuation differences on the right-hand side of (7) attains its zero (as does the boundary integral (8)). Otherwise only approximate equidistribution is achieved. But by minimizing the Least Squares norm we determine a mesh for which $\boldsymbol{\Phi}_k$ is approximately equalized over the region in this average sense. The result is true in any number of dimensions.

If $Q=\mathrm{diag}(q_m)$, $(m=1,\ldots,M)$, the weighted norms on the left- and right-hand side of (7) (which are minimized simultaneously over interior parameters) are

$$\sum_{m=1}^{M} q_m \sum_{k=1}^{N}(\boldsymbol{\Phi}_k)_m^{\mathrm{t}}(\boldsymbol{\Phi}_k)_m \quad\text{and}\quad \sum_{m=1}^{M} q_m \sum_{k=1}^{N}\sum_{l=1}^{N}(\boldsymbol{\Phi}_k-\boldsymbol{\Phi}_l)_m^{\mathrm{t}}(\boldsymbol{\Phi}_k-\boldsymbol{\Phi}_l)_m \tag{9}$$

5.2. Function equidistribution

We now digress to consider the problem of function approximation in multidimensions, which can also be analyzed using this approach.

The well-known equidistribution principle in one dimension involves locating mesh points such that a measure of an underlying given scalar function $u(x)$ is equalized (or equidistributed) over each cell [16, 17]. A function

$$\phi_k = \int_{X_{j-1}}^{X_j} m(x)\,\mathrm{d}x \tag{10}$$

is defined as the desired measure of the solution u to be equidistributed, where $m(x)$ is known as the monitor function and depends on $u(x)$. The mesh points X_j are found by solving $\forall j$ the integral equality

$$\int_{X_{j-1}}^{X_j} m(x)\,\mathrm{d}x = \int_{X_j}^{X_{j+1}} m(x)\,\mathrm{d}x \tag{11}$$

For example, if the monitor function is the derivative, $m(x) = \mathrm{d}u/\mathrm{d}x$ (assumed positive), then (11) becomes

$$u(X_{j+1}) - u(X_j) = \int_{X_j}^{X_{j+1}} \frac{\mathrm{d}u}{\mathrm{d}x}\mathrm{d}x = \int_{X_{j-1}}^{X_j} \frac{\mathrm{d}u}{\mathrm{d}x}\mathrm{d}x = u(X_j) - u(X_{j-1}) \tag{12}$$

and then the mesh is determined by equal intercepts on the u-axis. This may also be written as

$$\left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)_{j-\theta_1}(X_j - X_{j-1}) = \left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)_{j+\theta_2}(X_{j+1} - X_j) \tag{13}$$

using mean values. Similarly, if the monitor function $m(x)$ is the derivative of the arclength s of the underlying function $u, m(x) = \mathrm{d}s/\mathrm{d}x$ and the mesh is determined by equal arcs on the solution manifold.

Equidistribution is a one-dimensional concept. However, a higher dimensional generalization of (11) may be defined in terms of a vector-valued monitor function $\underline{m}(\underline{x})$ as

$$\int_{\Omega_1} \underline{m}(\underline{x})\,\mathrm{d}\Omega = \int_{\Omega_2} \underline{m}(\underline{x})\,\mathrm{d}\Omega \tag{14}$$

corresponding to the vector-valued equidistributed quantity (cf. (10))

$$\underline{\phi}_k = -\int_{\Omega_k} \underline{m}(\underline{x})\,\mathrm{d}\Omega \tag{15}$$

Introducing an equidistributing function $e(x)$ for which $\underline{m}(\underline{x}) = \nabla e(\underline{x})$, (14) becomes

$$\int_{\Omega_1} \nabla e(\underline{x})\,\mathrm{d}\Omega = \int_{\Omega_2} \nabla e(\underline{x})\,\mathrm{d}\Omega \tag{16}$$

so that, using a form of Gauss' Theorem, $\underline{\phi}_k$ can be written (cf. (11))

$$\underline{\phi}_k = \oint_{\partial\Omega_1} e(\underline{x})\hat{\underline{n}}\,\mathrm{d}\sigma = \oint_{\partial\Omega_2} e(\underline{x})\hat{\underline{n}}\,\mathrm{d}\sigma \tag{17}$$

From (17),

$$\sum_k \underline{\phi}_k = -\sum_k \int_{T_k} \underline{m}(\underline{x})\,\mathrm{d}\Omega = -\sum_k \int_{T_k} \nabla e(\underline{x})\,\mathrm{d}\Omega = \sum_k \oint_{\partial T_k} e(\underline{x})\hat{\underline{n}}\,\mathrm{d}\sigma = \oint_{\partial\Omega} e(\underline{x})\hat{\underline{n}}\,\mathrm{d}\sigma \tag{18}$$

through cancellation over the internal edges of the mesh, so we have that $\sum_k \underline{\phi}_k$ is independent of interior parameters.

Two examples of the equidistributing function $e(\underline{x})$ are the underlying function $u(\underline{x})$ itself and the arclength distance on the u manifold taken in the direction of ∇u.

Now let $e(\underline{x})$ be approximated by a *continuous* piecewise linear function E with vertex values $e(\underline{X})$ so that ∇E is constant in each triangle. Then we can define a discretized

equidistributed quantity (cf. (15) and (16))

$$\underline{\mathbf{\Phi}}_k = \oint_{\partial T_k} E\hat{\underline{n}}\,\mathrm{d}\sigma = -\int_{T_k} \nabla E\,\mathrm{d}S \tag{19}$$

which also satisfies the cancellation property over internal edges since

$$\sum_k \underline{\mathbf{\Phi}}_k = -\sum_k \int_{T_k} \nabla E\,\mathrm{d}S = \sum_k \oint_{\partial T_k} E\hat{\underline{n}}\,\mathrm{d}\sigma = \oint_{\partial\Omega} E\hat{\underline{n}}\,\mathrm{d}\sigma \tag{20}$$

(cf. (18)), E being continuous.

Since E is assumed to be piecewise linear, then $\underline{\mathbf{\Phi}}_k$ in (19) can be written in either of the two forms

$$\underline{\mathbf{\Phi}}_k = \left(\sum_{v=1}^{3} \frac{1}{2}(E(\underline{X}_{v1}) + E(\underline{X}_{v2}))\underline{N}_v \right)_k = -S_k(\nabla E)_k \tag{21}$$

where $\underline{N}_v = (-(\Delta Y)_v, (\Delta X)_v)^{\mathrm{t}}$ is again the normal to the side opposite the vertex v, measured inward, having the length of that side (see Figure 1), with the property $\sum \underline{N}_v = 0$. The suffices $v1, v2$ again indicate the vertices of the triangle other than v. The first of (21) then gives

$$\underline{\mathbf{\Phi}}_k = -\frac{1}{2}\left(\sum_{v=1}^{3} E(\underline{X}_v)\underline{N}_v \right)_k = \frac{1}{2}\left(\sum_{v=1}^{3} (\Delta E)_v \begin{pmatrix} -Y \\ X \end{pmatrix}_v \right)_k \tag{22}$$

using (5) and summation by parts.

Two examples of the equidistributing function E are a piecewise linear approximation U to the underlying function u and an arclength equidistributing function on the manifold of U in the direction of the gradient ∇U.

From the identity (7) with $\mathbf{\Phi} = \underline{\mathbf{\Phi}}$, $Q = I$, since $\sum_k \underline{\mathbf{\Phi}}_k$ is independent of interior points, the l_2 norm of $\underline{\mathbf{\Phi}}_k$ and the l_2 norm of the differences in $\underline{\mathbf{\Phi}}_k$ are again minimized simultaneously over interior parameters. Therefore, by minimizing $\sum_k \underline{\mathbf{\Phi}}_k^{\mathrm{t}} \underline{\mathbf{\Phi}}_k$ we may obtain an optimal mesh over which $\underline{\mathbf{\Phi}}_k$ is equidistributed in this least squares sense.

6. MINIMIZING THE LEAST SQUARES NORMS

Although in principle the least squares norms on the left-hand side of (7) may be minimized over any interior parameters (including the parameters of the solution) we shall concentrate on minimizing with respect to nodal positions $\underline{X}$, relying on a separate calculation using the iterative steps of multidimensional upwinding for the solution. These iterative solution steps do not attempt to decrease the l_2 norm but rather rely on progressing to the solution through physical states (see also Reference [18]).

In order to carry out the minimization of the least squares norm we may either use a descent method or alternatively solve the normal equations directly. An advantage of the former is that the functional is always reduced in an iterative step and can be monitored. Either way the gradient of the functional is required. We shall treat the two cases of the previous section in reverse order. We use a Lagrangian description in which $\underline{\mathbf{F}}$ or E depends on the $\underline{X}$'s.

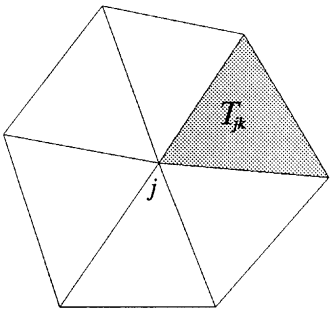

Figure 2. A local patch of elements surrounding node j.

6.1. The function approximation case

In order to calculate the gradient of $\sum_k \underline{\mathbf{\Phi}}_k^{\mathrm{t}} \underline{\mathbf{\Phi}}_k$ with respect to the nodal variable $\underline{X}_j$ we use the representations in (21) and (22) to obtain

$$\begin{aligned}\nabla_{\underline{X}_j} \sum_k \underline{\mathbf{\Phi}}_k^{\mathrm{t}} \underline{\mathbf{\Phi}}_k &= 2 \sum_{jk} (\nabla_{\underline{X}_j} \underline{\mathbf{\Phi}}_{jk}^{\mathrm{t}}) \underline{\mathbf{\Phi}}_{jk} \\ &= \sum_{jk} \left\{ -(\underline{N}_{jkj}^{\mathrm{t}} (\nabla_{\underline{X}_j} E(\underline{X}_j))) \underline{\mathbf{\Phi}}_{jk} + (\Delta E)_{jkj} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \underline{\mathbf{\Phi}}_{jk} \right\} \end{aligned} \tag{23}$$

using (5), where j is any node of the triangulation, jk runs over the triangles surrounding node j, and jkj denotes the side of the triangle T_{jk} opposite node j (see Figure 2).

Since in triangle jk

$$(\Delta E)_{jkj} = (\nabla E)_{jk}^{\mathrm{t}} \begin{pmatrix} \Delta X \\ \Delta Y \end{pmatrix}_{jkj} = (\nabla E)_{jk}^{\mathrm{t}} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \underline{N}_{jkj} \tag{24}$$

the gradient (23) becomes

$$\nabla_{\underline{X}_j} \sum_k \underline{\mathbf{\Phi}}_{jk}^{\mathrm{t}} \underline{\mathbf{\Phi}}_{jk} = \sum_{jk} \left\{ -\underline{N}_{jkj}^{\mathrm{t}} (\nabla_{\underline{X}_j} E(\underline{X}_j)) - (\nabla E)_{jk}^{\mathrm{t}} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \underline{N}_{jkj} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \right\} \underline{\mathbf{\Phi}}_{jk} \tag{25}$$

Using the definition of $\underline{\mathbf{\Phi}}_{jk}$ from the second of (21) and the fact that

$$\nabla_{\underline{X}_j} (E(\underline{X}_j)) = \nabla e(\underline{x})|_{\underline{x} = \underline{X}_j} \tag{26}$$

(since E is the interpolant of e) we obtain from (25)

$$\nabla_{\underline{X}_j} \sum_k \underline{\mathbf{\Phi}}_k^{\mathrm{t}} \underline{\mathbf{\Phi}}_k = \sum_{jk} S_{jk} \{ \underline{N}_{jkj}^{\mathrm{t}} \nabla e|_{\underline{x} = \underline{X}_j} + \underline{N}_{jkj}^{\mathrm{t}} (\nabla E)_{jk}^{\perp} \} (\nabla E)_{jk} \tag{27}$$

where

$$(\nabla E)_{jk}^{\perp} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} (\nabla E)_{jk} \tag{28}$$

Finally, approximating $\nabla e|_{\underline{x}=\underline{X}_j}$ by $(\nabla E)_{jk}$ in each element jk, (27) reduces to

$$\nabla_{\underline{X}_j} \sum_k \underline{\mathbf{\Phi}}_k^{\mathrm{t}} \underline{\mathbf{\Phi}}_k \approx \sum_{jk} S_{jk} (\nabla E)_{jk}^2 \underline{N}_{jkj} \tag{29}$$

An alternative expression for (29) is as follows. Let L_{jkj} be the length of the side of the triangle T_{jk} opposite node j (see Figure 2) and $\underline{X}_{jkj}$ be the foot of the perpendicular from $\underline{X}_j$ to that side. Then (29) becomes

$$\nabla_{\underline{X}_j} \sum_k \underline{\mathbf{\Phi}}_k^{\mathrm{t}} \underline{\mathbf{\Phi}}_k \approx \frac{1}{2} \sum_{jk} L_{jkj}^2 (\nabla E)_{jk}^2 (\underline{X}_j - \underline{X}_{jkj}) = \sum_{jk} w_{jkj} (\underline{X}_j - \underline{X}_{jkj}) \tag{30}$$

where

$$w_{jkj} = \tfrac{1}{2} L_{jkj}^2 (\nabla E)_{jk}^2 \tag{31}$$

An approximate steepest descent method for the minimization is therefore,

$$\underline{X}_j^{\mathrm{new}} = \underline{X}_j - \tau_j \sum_{jk} w_{jkj} (\underline{X}_j - \underline{X}_{jkj}) \tag{32}$$

where τ_j is a positive relaxation parameter. Since the w_{jkj}'s are positive, then with an appropriate choice of τ_j we obtain the iteration

$$\underline{X}_j^{\mathrm{new}} = \underline{X}_j - \gamma_j \frac{\sum_{jk} w_{jkj} (\underline{X}_j - \underline{X}_{jkj})}{\sum_{jk} w_{jkj}} = (1 - \gamma_j) \underline{X}_j + \gamma_j \frac{\sum_{jk} w_{jkj} \underline{X}_{jkj}}{\sum_{jk} w_{jkj}} \tag{33}$$

where γ_j is another positive relaxation factor.

Although the right-hand side of (33) is a convex function of the $\underline{X}$'s if $\gamma_j < 1$, there are still examples of meshes in which mesh tangling can occur. To avoid tangling it is necessary to introduce a limiter [19, 20].

If $\gamma_j < \frac{1}{2}$ the iteration matrix for (33) (written as a system) has norm < 1 giving a convergent iteration, although convergence can be slow. Because of the inherent non-linearity only a local minimum can be expected. The limit satisfies the normal equations (or equivalently the approximate gradient is zero) when the $\underline{X}$'s satisfy

$$\sum_{jk} w_{jkj} (\underline{X}_j - \underline{X}_{jkj}) = 0, \quad \text{i.e.} \quad \underline{X}_j = \frac{\sum_{jk} w_{jkj} \underline{X}_{jkj}}{\sum_{jk} w_{jkj}} \tag{34}$$

$\forall j$ (cf. (13)). Thus $\underline{X}_j$ is a convex average of the position vectors of the intersections of the normals with the opposite side in the surrounding triangles.

The w's depend on the underlying function $u(\underline{x})$ through the dependence of ∇E on ∇U. Examples of discrete equidistributing functions are $E = U, M = \nabla U$ and $E = S, M = \nabla S = \sqrt{(1 + \nabla U^2)}$ where S is the arclength on the discretised manifold measured in this direction. The corresponding w's are (cf. (31))

$$w_{jkj} = \tfrac{1}{2} L_{jkj}^2 (\nabla U)_{jk}^2 \tag{35}$$

and

$$w_{jkj} = \tfrac{1}{2} L_{jkj}^2 (1 + (\nabla U)_{jk}^2) \tag{36}$$

The first of these satisfies the internal cancellation property (20) exactly but the second only approximately.

6.2. *The fluctuation*

The gradient of the weighted l_2 norm (9) of the fluctuation $\mathbf{\Phi}_k$ of (4) is

$$\nabla_{\underline{X}_j} \sum_{m=1}^{M} q_m \sum_{k=1}^{N} (\mathbf{\Phi}_k)_m^{\mathrm{t}} (\mathbf{\Phi}_k)_m = 2 \sum_{m=1}^{M} q_m \sum_{jk} (\nabla_{\underline{X}_j} (\mathbf{\Phi}_{jk}^{\mathrm{t}})_m)(\mathbf{\Phi}_{jk})_m \tag{37}$$

Using the two representations in (6) (dropping the suffix m for convenience)

$$\begin{aligned} &2\sum_{jk} (\nabla_{\underline{X}_j} \mathbf{\Phi}_{jk}^{\mathrm{t}}) \mathbf{\Phi}_{jk} \\ &\quad = \sum_{jk} \left\{ -(\nabla_{\underline{X}_j} \underline{\mathbf{F}}(\mathbf{U}(\underline{X}_j)))^{\mathrm{t}} \underline{N}_{jkj} \mathbf{\Phi}_{jk} + \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}^{\mathrm{t}} (\Delta \underline{\mathbf{F}}(\mathbf{U}(\underline{X})))_{jkj}^{\mathrm{t}} \mathbf{\Phi}_{jk} \right\} \end{aligned} \tag{38}$$

Since $\underline{\mathbf{F}}$ is piecewise linear in triangle jk

$$(\Delta \underline{\mathbf{F}})_{jkj} = (\nabla \underline{\mathbf{F}})_{jk}^{\mathrm{t}} \begin{pmatrix} \Delta X \\ \Delta Y \end{pmatrix}_{jkj} = (\nabla \underline{\mathbf{F}})_{jk}^{\mathrm{t}} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \underline{N}_{jkj} \tag{39}$$

and

$$\nabla_{\underline{X}_j} \underline{\mathbf{F}}(\mathbf{U}(\underline{X}_j)) = \nabla \underline{\mathbf{f}}|_{\underline{x} = \underline{X}_j}, \quad \mathbf{\Phi}_{jk} = -S_{jk} (\operatorname{div} \underline{\mathbf{F}})_{jk} \tag{40}$$

the right-hand side of (38) becomes

$$\sum_{jk} \left\{ -(\nabla \underline{\mathbf{f}})|_{\underline{x} = \underline{X}_j})^{\mathrm{t}} \underline{N}_{jkj} \underline{\mathbf{\Phi}}_{jk} + \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}^{\mathrm{t}} (\nabla \underline{\mathbf{F}}(\mathbf{U}(\underline{X})))_{jk}^{\mathrm{t}} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \underline{N}_{jkj} \mathbf{\Phi}_{jk} \right\} \tag{41}$$

leading to

$$\nabla_{\underline{X}_j} \sum_{k} \mathbf{\Phi}_k^{\mathrm{t}} \mathbf{\Phi}_k = \sum_{jk} S_{jk} \{ \underline{N}_{jkj}^{\mathrm{t}} (\nabla \underline{\mathbf{f}})|_{\underline{x} = \underline{X}_j} + (\nabla \underline{\mathbf{F}})_{jk}^{\perp} \underline{N}_{jkj}^{\perp} \} (\operatorname{div} \underline{\mathbf{F}}(\mathbf{U}(\underline{X})))_{jk} \tag{42}$$

where

$$\nabla \underline{\mathbf{F}}^{\perp} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \nabla \underline{\mathbf{F}} \tag{43}$$

and

$$\underline{N}_{jkj}^{\perp} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \underline{N}_{jkj} \tag{44}$$

Approximating $\nabla \underline{\mathbf{f}}$ by $(\nabla \underline{\mathbf{F}})_{jk}$ in triangle jk we can derive an approximate steepest descent step for the minimization of the l_2 norm of the form

$$\underline{X}_j^{\mathrm{new}} = \underline{X}_j - \tau_j \sum_{m=1}^{M} q_m \sum_{jk} S_{jk} \{ (\nabla \underline{\mathbf{F}}(\mathbf{U}(\underline{X})))_{jk}^{\mathrm{t}} \underline{N}_{jkj} + (\nabla \underline{\mathbf{F}}(\mathbf{U}(\underline{X})))_{jk}^{\perp} \underline{N}_{jkj}^{\perp} \}_m \{ (\operatorname{div} \underline{\mathbf{F}})_{jk} \}_m \tag{45}$$

where τ_j is a positive relaxation factor.

The first term in the curly brackets in (45) arises from the variation with $\underline{X}$ within $\underline{\mathbf{F}}(\mathbf{U}(\underline{X}))$, while the second term comes from variation of the triangle areas. If $\mathbf{U}$ is regarded as independent of the mesh locations the first term is absent.

The choice of τ_j is less obvious than in the previous section. A standard approach is to use a line search or local quadratic model, and significantly there is an obvious quadratic model to be had here obtained by *lagging* the values of $\mathbf{U}$, in the sense that these values are temporarily frozen but updated at the next step. This is equivalent to suppressing variations over $\mathbf{U}$. The least squares norm becomes quadratic in $\underline{X}$ and there is a unique minimizer which is easily found [21]. Similarly, the normal equations are linear and the full range of linear solvers is accessible. However, the l_2 norm may not be reduced in this approach if the variations which are excluded are not independent of the variations which are included.

To illustrate the roles of the terms in (45) consider the scalar advection equation

$$\underline{a}.\nabla u = au_x + bu_y = 0 \tag{46}$$

with constant $\underline{a} = (a, b)^{\mathrm{t}}$ in each triangle, for which $\underline{\mathbf{F}}$ is the scalar $U\underline{a}$ and $\operatorname{div}\underline{\mathbf{F}}$ is $\underline{a}^{\mathrm{t}}\nabla U$. The gradient of $\underline{\mathbf{F}}$ is

$$\nabla\underline{\mathbf{F}} = \begin{pmatrix} aU_x & bU_x \\ aU_y & bU_y \end{pmatrix} = (\nabla U)\underline{a}^{\mathrm{t}} \tag{47}$$

Then the approximate steepest descent method (45) (with the q's taken to be 1) reduces to

$$\underline{X}_j^{\mathrm{new}} = \underline{X}_j - \tau_j \sum_{jk} S_{jk}(\underline{a}^{\mathrm{t}}\nabla U)_{jk}\{(\underline{a}_{jk}^{\mathrm{t}}\underline{N}_{jkj})(\nabla U)_{jk} + ((\nabla U)_{jkj}^{\mathrm{t}}\underline{N}_{jkj}^{\perp})\underline{a}_{jk}^{\perp}\} \tag{48}$$

where $\underline{a}^{\perp} = (b, -a)$ and τ_j is a positive relaxation factor (cf. References [6, 22]). When $\underline{a}$ varies with u but is lagged in the iteration or when U is treated as independent of $\underline{X}$, the first term in the curly brackets in (48) vanishes and the node movement is at right angles to $\underline{a}$ [21, 6, 22].

In this scalar case the mesh movement drives the fluctuations down almost to zero and the mesh is very nearly characteristic [8].

7. USE OF DEGENERATE TRIANGLES

In the presence of shocks or contact discontinuities least squares methods give inaccurate solutions which are unacceptable. One way to combat this problem is to subdivide the region and introduce degenerate (vertical) triangles at the interface [22]. For degenerate triangles we can still define the fluctuation $\mathbf{\Phi}_k$ (see (3)), which vanishes when the Rankine–Hugoniot jump conditions are satisfied. Then by minimizing the least squares norm of $\mathbf{\Phi}_k$ with moving nodes the position of the discontinuity can be adjusted, as in shock fitting methods, to approximately satisfy these conditions [11, 21].

An approximate solution is first obtained by the use of a multidimensional upwinding shock capturing scheme [11]. An initial discontinuous solution is then constructed by introducing degenerate triangles into the regions identified as containing a shock, using a shock identification technique. (This step may be carried out manually or the degenerate triangles can be added automatically using techniques that exist in the shock fitting literature—see for example Reference [23]. The position of the discontinuity so formed can then be improved by

minimizing the l_2 norm of $\mathbf{\Phi}$ with respect to node positions. When updating coincident nodal positions on the shock they must have the same update (so that the cell remains degenerate). This is obtained by minimization with respect to their common position.

The procedure may be interleaved with a descent least squares method for the *solution*, which is smooth on either side of the discontinuity, allowing the descent method to be used for the solution as well as the mesh.

8. IMPLEMENTATION

8.1. Function approximation

In the case of function equidistribution the mesh may be moved as in (33). A possible algorithm for the implementation of this procedure is:

8.1.1. Interleaving algorithm 1

(1) Set up an initial mesh.
(2) Sample the function or carry out a least squares best fit.
(3) Adjust the mesh using the updating formula (33).
(4) Repeat from (2) until convergence.

Results from the algorithm are not shown here, being very similar to those found in Reference [19].

8.2. Conservation laws

Where the weights (31) depend on the approximate solution of a conservation law or other PDE, a simultaneous minimization of the l_2 norm over *all* interior parameters, both mesh and solution values, are possible. In such an approach, which is discussed elsewhere in this volume (see also Reference [8]), the $\mathbf{U}$ and $\underline{X}$ values are taken to be independent, which contrasts with the function equidistribution approach in which an integral assumption is that $\mathbf{U}$ depends on the $\{\underline{X}\}$'s.

However, it is not necessary to carry out the minimization over both solution and mesh values. The steepest descent step (33) for the mesh may be interleaved with any solution procedure, allowing a full range of choice of methods, although monotonicity of the functional is sacrificed when using other methods for the solution. The descent steps may be incorporated in an overall iterative algorithm for both the mesh and the solution in which iterative solution updates are alternated with mesh movement descent steps. The approach can then be applied to the solution of any PDE. This suggest the algorithm:

8.2.1. Interleaving algorithm 2

(1) Set up an initial mesh.
(2) Obtain an approximate solution of the conservation law on this mesh using a multidimensional upwinding scheme.
(3) Carry out one step of an iteration procedure for the solution of the conservation law.
(4) Adjust the mesh using the updating formula (45) with q_m chosen to be 1 or to pick out the first component (e.g. density in the Euler equations).

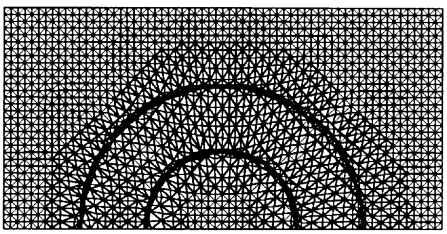

Figure 3. Circular advection with mesh movement.

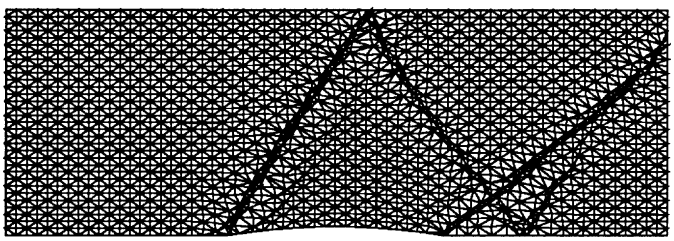

Figure 4. Adapted mesh for the Euler flow through a constricted channel.

(5) Repeat from (3) until the mesh has converged.
(6) Repeat from (2) on this mesh until the solution has also converged.

Two examples are shown in Figures 3 and 4 below.

When using degenerate triangles the algorithm is:

8.2.2. Interleaving algorithm 3

(1) Set up an initial mesh.
(2) Obtain an initial approximate solution using a multidimensional upwinding shock capturing scheme.
(3) Construct an initial discontinuous solution by introducing degenerate triangles into regions identified as containing shocks.
(4) Improve the position of the discontinuity by minimizing the l_2 norm of $\mathbf{\Phi}$ with respect to node positions.
(5) Improve the solution by minimizing the l_2 norm of $\mathbf{\Phi}$ with respect to the solution values.
(6) Repeat from (2) until both mesh and solution have converged.

An example is shown in Figure 5 below. Note that step (2) in algorithms 2 and 3 ensures that the inflow and outflow values of the boundary integral (8) balance, giving near equality between the l_2 and equidistribution norms. This is in contrast to the function approximation case.

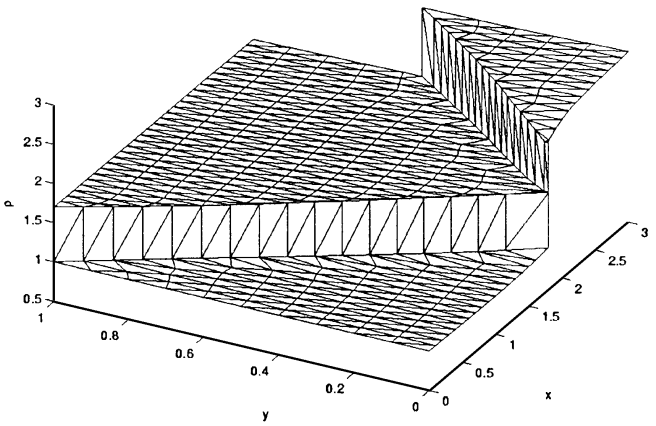

Figure 5. Solution with degenerate triangles for the Yee shock reflection problem.

8.3. *Local sweeps*

The sequence of operations in each of these interleaving algorithms can be carried out on a purely local basis [3]. In applying (45) we in effect carry out a single local step of an inner iteration for each equation. Such local steps involve only one node at a time and are cheap and easy to implement. The iteration then continues by sweeping through the mesh repeatedly. By interleaving local mesh update steps with local solution update steps the iteration can proceed on a local basis, one node at a time. The sweeps through the mesh may be carried out in either a Jacobi or Gauss–Seidel manner.

In effect a sequence of local problems in solved. Each mesh movement step is equivalent to applying the minimizations of Section 5.1 to a local patch of triangles and reduces the local l_2 norm.

In Algorithm 3 the inflow boundary values of the *solution* are overwritten at the end of each sweep through the mesh, which violates the steepest descent monotonicity property of the functional. However, this situation does not arise in the case of least squares *mesh* updates.

9. NUMERICAL RESULTS

We show results for a scalar case and an Euler case using Algorithm 2. The scalar case is that of circular advection. Consider the scalar two-dimensional advection equation

$$\underline{a}(\underline{x}).\nabla u = 0 \tag{49}$$

where $\underline{a}(\underline{x}) = (y, -x)$ in a rectangle $-1 \leqslant x \leqslant 1, 0 \leqslant y \leqslant 1$, which generates a semicircular hump swept out by the initial data, here chosen to be

$$U = \begin{cases} 1 & -0.6 \leqslant x \leqslant -0.5 \\ 0 & \text{otherwise} \end{cases} \tag{50}$$

Results are shown in Figure 3 and nodes are seen to cluster towards the discontinuity under the action of arclength equidistribution in the direction of the gradient of U. This is in contrast

to the result in References [6, 8], using only one of the terms of (48), where only nodes adjacent to the discontinuity move.

The Euler case is that of flow in a channel with a 4% circular constriction on its lower surface [10, 20]. The freestream Mach number is 1.4 and the initial calculation is carried out on a regular 4096 mesh. The iterative strategy is an in Algorithm 2. The mesh movement is run for 500 iterations after the functional has dropped by four orders of magnitude. The overall number of time steps is such that the adaptation has a negligible effect on the average expense per step. The final mesh is shown in Figure 4.

For an illustration of this use of degenerate triangles a result is shown for the Euler equations [21, 11]. The example exhibits the shock fitting capabilities of the method for a purely supersonic flow which has an exact solution [24].

The computational domain is of length 3 and width 1. Supersonic inflow boundary conditions, given by values $\mathbf{U} = (1.0, 2.9, 0.60)^t$ and $\mathbf{U} = (1.7, 4.453, -0.86, 9.87)^t$ of the conservative variables, are imposed on the left and upper boundaries, respectively. At the right-hand boundary supersonic outflow conditions are applied, while the lower boundary is treated as a solid wall.

The boundary conditions are chosen so that the shock enters the top left hand corner at an angle of 29° to the horizontal and is reflected by a flat plate on the lower boundary. The flow in regions away from shocks is constant.

The figure (taken from Reference [11]) shows a shock reflection problem in which a solution obtained by a shock capturing multidimensional upwinding scheme has been improved by adjusting the mesh using least squares fluctuation minimization with degenerate cells at the shocks. The results are shown in Figure 4 where the density contours are plotted. The shock comes in from the top left-hand corner at an angle of 29.2° to the horizontal and the solution is virtually constant apart from the discontinuities, in close agreement with the analytic solution.

10. CONCLUSIONS

In this paper we have considered fluctuations and monitors defined on triangles which satisfy an integral cancellation property over the internal edges of an unstructured mesh. For such quantities the l_2 norm and the l_2 norm of the differences are minimized simultaneously. Particular examples are the fluctuation associated with a system of conservation laws and a vector-valued equidistributed function which can be used for multidimensional function approximation.

The minimizations are carried out by a steepest descent approach which, in the function approximation case, reduces to a simple averaging formula. For the fluctuation the relaxation parameter associated with the steepest descent method may be chosen using a local quadratic model which arises naturally when the solution variables are frozen.

The fluctuation used here, which is continuous across triangle edges, may be used with degenerate triangles to position shocks [22]. By minimizing the fluctuation in a degenerate triangle, which is a measure of satisfaction of the jump condition, an approximate position of the shock can be found which may be manoeuvred into an accurate position using node movement. The least squares minimization is carried out with respect to interior solution parameters on either side of the shock, which gives a good approximation of the adjacent smooth regions of the flow.

Results from these mesh movement strategies, used in conjunction with a multidimensional upwinding method, have been shown for three steady-state test cases, those of scalar circular advection, gas flow through a constricted channel, and a shock reflection problem illustrating the use of degenerate triangles.

Further developments include the extension to time dependent problems. In one approach the mesh moves so that for example a conserved quantity remains invariant [25]. In another approach [2] a steepest descent method has been used to generate mesh movement in the context of a Stefan problem. It may be possible to develop the averaging formula (33) in the same way as is done for MMPDEs in Reference [17]. Finally, the moving finite element (MFE) method [26], which is known to be optimal for certain steady state problems [4], has the useful property that for scalar hyperbolic equations nodes move along the characteristics, asymptotically carrying L_2 best fits [27].

ACKNOWLEDGEMENTS

The author wishes to thank Phil Roe of the University of Michigan for useful discussions while visiting the University of Reading under an EPSRC Visiting Fellowship. Grateful thanks are also due to Matthew Hubbard and Stephan Leary for a fruitful collaboration in this area and for the numerical results.

REFERENCES

1. Giles MB. On adjoint equations for error analysis and optimal grid adaptation in CFD. *Report* 97/11, Oxford University Computing Laboratory, 1997.
2. Beckett G *et al*. A moving mesh finite element method for the solution of two-dimensional Stefan problems. *Journal of Computational Physics* 2002, to appear.
3. Tourigny Y, Hulsemann F. A new moving mesh algorithm for the finite element solution of variational problems. *SIAM Journal of Numerical Analysis* 1998; **34**:1416–1438.
4. Jimack PK. Local minimization of errors and residuals using the moving finite element method. *Report* 98.17, School of Computer Science, University of Leeds, 1998.
5. Baines MJ. Moving finite element, least squares, and finite volume approximations of steady and time-dependent PDEs in multidimensions. *Journal of Computational and Applied Mathematics* 2001; **128**:363–382.
6. Roe PL. Compounded of many simples: In *Proceedings of Workshop on Barriers and Challenges in CFD*, *ICASE*, *NASA Langley*, Ventakrishnan *et al*. (eds). Kluwer: Dordrecht, 1998; 241–258.
7. Miller K, Baines MJ. Least squares moving finite elements. *IMA Journal of Numerical Analysis* 2001, to appear.
8. Baines MJ, Leary SJ. Fluctuation and signals for scalar hyperbolic equations on adjustable meshes. *Communications in Numerical Methods in Engineering* 1999; **15**:877–886.
9. Fazlo R, LeVeque RJ. Moving grid methods for hyperbolic conservation laws using CLAWPACK. *Computational Mathematics and Applications* 2002, to appear.
10. Baines MJ, Hubbard ME. Multidimensional upwinding with grid adaptation. In *Numerical Methods for Wave Propagation*, Toro EF, Clarke JF (eds). Kluwer: Dordrecht, 1998.
11. Baines MJ *et al*. Multidimensional least squares fluctuation schemes with adaptive mesh movement for steady hyperbolic equations. *SIAM Journal of Science and Computing* 2002, to appear.
12. Baines MJ, Leasts squares and approximate equidistribution in multidimensions. *Numerical Methods for Partial Differential Equations* 1999; **15**:605–615.
13. Roe PL. Fluctuation and signals: a framework for numerical evolution problems. In *Proceedings of IMA Conference on Numerical Methods for Fluid Dynamics*, Morton KW, Baines MJ (eds). Academic Press: Reading, 1982.
14. Deconinck H *et al*. High resolution shock capturing cell vertex schemes for unstructured grids. *Computational Fluid Dynamics*. VKI Lecture Notes 1994-05, 1994.
15. Mesaros LM, Roe PL. Multidimensional fluctuation splitting schemes based on decomposition methods. *Proceedings of the 12th AIAA CFD Conference*, San Diego, 1995.
16. Dorfi EA, Drury LO'C. Simple Adaptive Grids for 1D Initial Value Problems. *Journal of Computational Physics* 1987; **69**:175–195.

17. Huang W, Russell RD. Adaptative mesh movement—the MMPDE approach and its applications. *Journal of Computational and Applied Mathematics* 2001; **128**:383–398.
18. Wood WA, Kleb WL. On multidimensional unstructured mesh adaption. AIAA paper 99-3254, *14th AIAA CFD Conference*, 1999.
19. Baines MJ. Grid adaptation via node movement. *Applied Numerical Methods* 1998; **26**:77–96.
20. Hubbard ME. Multidimensional upwinding and grid adaptation for conservation laws. *Ph.D. Thesis*, University of Reading, 1996.
21. Leary SJ. Least squares methods with adjustable nodes for steady hyperbolic PDEs. *Ph.D. Thesis*, University of Reading, 1999.
22. Roe PL. Fluctuation splitting schemes on optimal grids. AIAA paper 97-2304, 1997.
23. Trepanier JY *et al.* A conservative shock fitting method on unstructured grids. *Journal of Computational Physics* 1996; **126**:421–433.
24. Yee H *et al.* Implicit total variation diminishing (TVD) schemes for steady state calculations. *Journal of Computational Physics* 1985; **57**:327–366.
25. Budd CJ, Piggott MD. The geometrical integration of scale-invariant ordinary and partial differential equations. *Journal of Computational and Applied Mathematics* 2001; **128**:399–422.
26. Carlson NN, Millert K. Design and application of a gradient weighted moving finite element method, I: in one dimension, II: in two dimensions. *SIAM Journal of Scientific Computing* 1998; **19**:728–798.
27. Baines MJ. *Moving Finite Elements*. Oxford University Press: Oxford, 1994.

