A97-32492

AIAA-97-2034

FLUCTUATION SPLITTING SCHEMES ON OPTIMAL GRIDS

Philip L. Roe*

W. M. Keck Foundation Laboratory for Computational Fluid Dynamics

Department of Aerospace Engineering, The University of Michigan, Ann Arbor, MI 48109

Abstract

In this paper we extend the fluctuation-splitting method by allowing the nonvanishing fluctuations to drive the vertex placements as well as the vertex values. For the generalised Cauchy-Riemann (Prandtl-Glauert) equations, a simple least-squares minimisation carried out with respect to u, v, x, y as unknowns creates a grid responsive to the physics of the solution, if the objective function is correctly defined. We speculate on the proper generalization of this idea.

I. Introduction

The Fluctuation Splitting method[5,?] can be applied on any structured or unstructured grid on which the values of the conserved variables $\mathbf{U}$ are stored at the vertices. The first step is to compute on each cell the fluctuation, or integrated value of $\partial_t \mathbf{U}$.

$$\phi_T = \int_T \partial_t \mathbf{U}\, dA \tag{1}$$

Usually, this integral is carried out via some discrete quadrature rule, on the assumption that the integrand has some simple form. For example, suppose we are dealing with the Euler equations written in quasi-linear form

$$\partial_t \mathbf{U} + A\, \partial_x \mathbf{U} + B\, \partial_y \mathbf{U} = 0. \tag{2}$$

and that the grid is composed of triangles. The assumption that $\mathbf{U}$ itself varies linearly over each triangles leads to rather messy integral integrals involving logarithms, but the asumption that the " parameter vector"

$$\mathbf{W} = \sqrt{\rho}(1, u, v, h)^T$$

varies linearly gives rise to the simple formula

$$\phi_T = \frac{1}{2} \sum_{j=1}^{j=3} \left(A(\overline{W})\Delta y_j - B(\overline{W})\Delta x_j \right) \mathbf{W}_j \tag{3}$$

in which $\mathbf{W}_j$ is the value 0f $\mathbf{W}$ at node j, the elementwise constant matrices A and B are both evaluated at the state defined by

$$\overline{\mathbf{W}} = \frac{1}{3} \sum_j \mathbf{W}_j,$$

and $(\Delta x_j, \Delta y_j)$ is the side of the element opposite vertex j. These formulae generalise readily to three dimensions. However, my belief is that a strength of the FS approach is its flexibility; adherence to a consistent finite-element interpretation may not be necessary.

In whatever way the fluctuation has been calculated, the next step is to distribute it to the nodal values. That is to say, each node of the cell is updated according to

$$S_j \mathbf{U}_j = S_j \mathbf{U}_j + \Delta t \sum_T \alpha_{j_T} \phi_T, \tag{4}$$

where S_j is an area associated with node j, chosen to ensure conservation, and α_{j_T} is a coefficient, generally matrix-valued, that vanishes unless vertex j belongs to triangle T. Schemes of this form can be coded with very low communication costs as loops over all triangles, but can be brought to a surprising degree of sophistication. Appropriate rules for choosing α_{j_T} have been given in many places; they have been most completely developed for converging on the steady solution. They can be made to reflect the hyperbolic or elliptic nature of the differential problem. A convective character is given to the algorithm by insisting that

$$\sum_j \alpha_{j_T} = 1,$$

where the sum is over all vertices of triangle T. A dissipative (elliptic) character is imparted by the choice

$$\sum_j \alpha_{j_T} = 0.$$

In the first case, the method is conservative if Δt is a global constant; in the second case conservation is ensured for any local choice of Δt.

*Professor, Fellow, AIAA

Analysis of the vector Φ_T reveals much about the local nature of the flow. It is shown in[8] that projecting Φ_T into various subspaces in effect identifies events due to convection of entropy and enthalpy, which should be treated by convective methods, and events due to potential flow, which should be treated dissipatively, unless the flow is locally supersonic, when a bifurcation into two wavelike phenomena occurs. The main purpose of this paper is to indicate that this wealth of information can be used to adapt the grid also, by using nonvanishing fluctuations to drive movement of the nodal positions as well as changing the nodal values.

Although many research problems still need to be solved, there is the potential to create an automatic mesh movement scheme having the following features;

1. In regions dominated by hyperbolic behaviour, the mesh will align itself (approximately) with those characteristics carrying the information.

2. In regions dominated by elliptic behaviour, the mesh will attempt to smooth itself out, while paying attention to non-isotropic behaviour such as the mostly 'sideways' influence of flow at Mach numbers just below unity.

3. Conflict between the above items can be used to motivate point insertion.

4. Small fluctuations can be used to motivate point removal.

5. The process can be driven by a minimization principle that motivates diagonal swapping or the three-dimensional eqivalents.

The grids that result from such an approach are unlikely to resemble the grids produced by such classical methods as Delaunay triangulation. Although optimal in many respects for solving Laplace's equation, the Delaunay strategy essentially assumes that the solution being sought is as bland and featureless as solutions to Laplace's equation typically are. Directionality, anisotropy and local features cannot be captured by such an approach. The fact that grids need to be tailored to their applications has been known to practitioners for a loing time, but has only recently begun to receive theoretical treament.[3,1,6] See[10] for a related approach to scalar problems obeying a variational principle, and[2] for alternative motivations for mesh movement.

To give a preliminary treatment of the strategy we will fix attention below on the Cauchy-Riemann system, generalised by a 'compressibilty factor' that permits a transition to hyperbolic behaviour. The viewpoint taken will be one of geometric approximation, attempting to 'triangulate' a certain manifold in $I\!R^4$. This is significantly different from the view taken in,[?] which also contains a preliminary account of the mesh movement strategy. The final form that a successful algorithm would take remains rather open.

II. Generalized Cauchy-Riemann System

Consider the equations

$$(1-M^2)\partial_x u + \partial_y v = 0, \tag{5}$$

$$\partial_x v - \partial_y u = 0, \tag{6}$$

which govern inviscid, irrotational, slightly compressible, two dimensional flow. For $M = 0$ these are (with a notational change) the Cauchy-Riemann equations. For $M > 1$ they undergo a change of type and become hyperbolic rather than elliptic. This paper will explore a numerical technique that solves them in a unified way for any value of M.

It is well-known that we can exchange the roles of dependent and independent variables through the hodograph transformation

$$\partial_u x = \partial_y v/j, \qquad \partial_v x = -\partial_y u/j,$$

$$\partial_u y = -\partial_x v/j, \qquad \partial_v y = \partial_x u/j,$$

where

$$j = \partial_x u\, \partial_y v - \partial_y u\, \partial_x v = (\partial_v y\, \partial_u x - \partial_v x\, \partial_u y)^{-1} \tag{7}$$

is the Jacobian of the transformation. Evidently we have governing equations in the hodograph plane,

$$\partial_u x + (1-M^2)\partial_v y = 0, \tag{8}$$

$$\partial_u y - \partial_v x = 0. \tag{9}$$

Draw some triangle 123 in the physical plane and its image in the hodograph plane. Assume that both triangles are so small that variation is linear within each of them. Then the transformation j is constant within them and precise discrete analogs exist for all of the above results.

Specifically, introduce the vector $\mathbf{x} = (x_1, x_2, x_3)^T$ and $\mathbf{y}, \mathbf{u}, \mathbf{v}$ analogously, and the rank-two tensor P defined by

$$P = \frac{1}{2}\begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix} \tag{10}$$

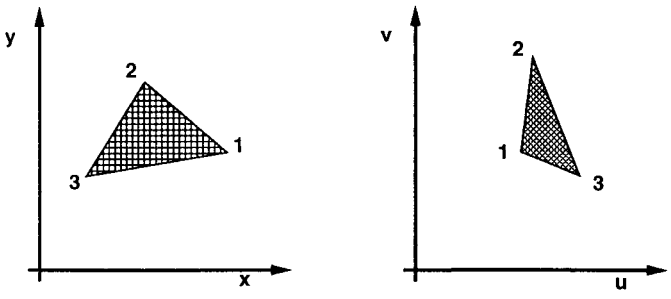


Figure 1: Small triangular regions of physical space (left) and velocity (hodograph) space (right).

Note that because P is antisymmetric, $P\mathbf{ab} = -P\mathbf{ba}$, and $d(P\mathbf{ab}) = P\mathbf{a}d\mathbf{b} - P\mathbf{b}d\mathbf{a}$. Then we have

$$\begin{aligned} U_X &= P\mathbf{uy} = P\mathbf{xy}\,\partial_x u = P\mathbf{uv}\,\partial_v y, \\ U_Y &= -P\mathbf{ux} = P\mathbf{xy}\,\partial_y u = -P\mathbf{uv}\,\partial_v x, \\ V_X &= P\mathbf{vy} = P\mathbf{xy}\,\partial_x v = -P\mathbf{uv}\,\partial_u y, \\ V_Y &= -P\mathbf{vx} = P\mathbf{xy}\,\partial_y v = P\mathbf{uv}\,\partial_u x, \end{aligned} \tag{11}$$

where $P\mathbf{xy}$, $P\mathbf{uv}$ are the areas of the triangles in the physical and hodograph planes respectively. These formulae are merely the usual finite-element approximations to the first derivatives. Discrete versions of the governing equations are

$$(1-M^2)P\mathbf{uy} - P\mathbf{vx} = 0 \tag{12}$$

$$P\mathbf{vy} + P\mathbf{ux} = 0 \tag{13}$$

A geometrical interpretation of the governing equations in this form is that they assert the equality or proportionality of the areas of each triangle when projected onto various directions in $\mathbb{R}^4$. From this viewpoint the governing equations define properties of a solution surface in the 4-space (u, v, x, y) and the discrete solution is some approximate triangulation of that surface.

An interesting quantity is

$$\begin{aligned} F &= \frac{1}{2}((1-M^2)U_X + V_Y)^2 + \frac{k^2}{2}(V_X - U_Y)^2 \\ &= \frac{1}{2}\left[\sum (1-M^2)(\mathbf{u}^T P\,\mathbf{y}) - \mathbf{v}^T P\,\mathbf{x}\right]^2 \\ &\quad + \frac{k^2}{2}\left[\mathbf{v}^T P\,\mathbf{y} + \mathbf{u}^T P\,\mathbf{x}\right]^2 \qquad (14) \\ &= \frac{1}{2}D^2 + \frac{k^2}{2}\Omega^2, \qquad (15) \end{aligned}$$

which has equal claim to represent the error in either the physical or hodograph equations. The choice of the weight k is left open for the time being.

III. The Subsonic Case

Given a set of triangles that tessellate a domain of interest, a possible numerical method to solve (5,6) would be to minimise the sum of F over all triangles, with respect to the nodal values of $\mathbf{U} = \{u, v, x, y\}$. Simultaneously solving in the physical and hodograph planes is, in effect, employing a particular kind of adaptive grid. At a minimum we have, writing the sum of the Fs as $\mathcal{F}$.

$$\frac{\partial \mathcal{F}}{\partial \mathbf{U}_j} = \sum_{T \in T_j} \frac{\partial F_T}{\partial \mathbf{U}_j} = \sum_{T \in T_i} \left[D_T \frac{\partial D_T}{\partial \mathbf{U}_j} + \Omega_T \frac{\partial \Omega_T}{\partial \mathbf{U}_j} \right] = 0. \tag{16}$$

The notation here is that j denotes a particular node, and T_j is one of the triangles sharing that node. Later we use j_T to mean a node belonging to triangle T. Each term in the sum can be read off by noting that the change in a given cell due to arbitrary changes at the vertices can be written

$$\begin{aligned} dF = & -[(1-M^2)D\mathbf{y}^T + k^2\Omega\mathbf{x}^T]P\ d\mathbf{u} \\ & +[D\mathbf{x}^T - k^2\Omega\mathbf{y}^T]P\ d\mathbf{v} \\ & -[D\mathbf{v}^T - k^2\Omega\mathbf{u}^T]P\ d\mathbf{x} \\ & +[(1-M^2)D\mathbf{u}^T + k^2\Omega\mathbf{v}^T]P\ d\mathbf{y} \end{aligned}$$

Hence, introducing the further notation that $\Delta x_T, \Delta y_T$ is the vector along the side of element T opposite to node j taken anticlockwise,

$$\begin{aligned} \frac{\partial \mathcal{F}}{\partial u_j} &= -(1-M^2)\sum_{T \in T_j} D_T \Delta y_T - k^2 \sum_{T \in T_j} \Omega_T \Delta x_T \\ &= -(1-M^2)^2 \sum_{T \in T_j} \Delta y_T \sum_{i \in j_T} u_i \Delta y_i \\ &\quad + (1-M^2)\sum_{T \in T_j} \Delta y_T \sum_{i \in j_T} v_i \Delta x_i \\ &\quad - k^2 \sum_{T \in T_j} \Delta x_T \sum_{i \in j_T} v_i \Delta y_i \\ &\quad - k^2 \sum_{T \in T_j} \Delta x_T \sum_{i \in j_T} u_i \Delta x_i \end{aligned} \tag{18}$$

and this expression will of course vanish at a minimum.

Consider the terms involving v_j. It is easy to show that

$$\sum_{T \in T_j} \Delta y_T \sum_{i \in j_T} v_i \Delta x_i = \sum_{T \in T_j} \Delta x_T \sum_{i \in j_T} v_i \Delta y_i. \tag{19}$$

because the coefficient of v_i in each expression is twice the area of the quadrilateral $V_{i-1}V_iV_{i+1}V_j$. This being so, the choice of norm

$$k^2 = 1 - M^2$$

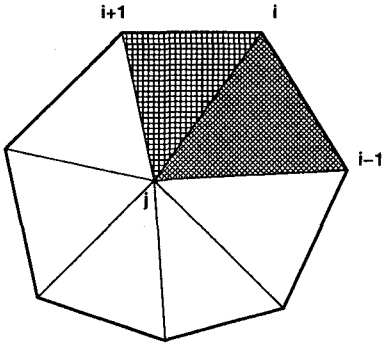


Figure 2: Triangulation around node j.

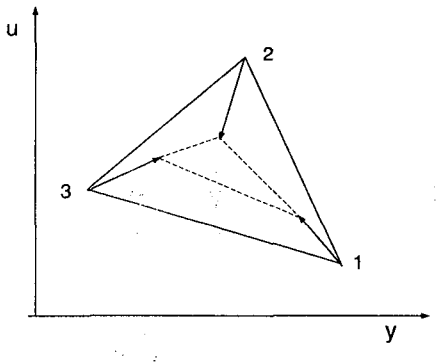


Figure 3: Node movement in the y, u plane due to nonvanishing D.

has the property of eliminating v from the equation satisfied at a minimum with respect to u. It is easy to verify that it also eliminates u from the equation that minimizes v; similarly y from the equation for x and x from the equation for y. In this norm uniquely, mimimizing the residual of the first-order system leads to discretizations of the four scalar equations

$$(1-M^2)\partial_{xx}(u,v) + \partial_{yy}(u,v) = 0 \quad (20)$$
$$\partial_{uu}(x,y) + (1-M^2)\partial_{vv}(x,y) = 0 \quad (21)$$

Moreover, when the Mach number vanishes, the discretization is the standard (Galerkin, second-order) finite-element method in both the physical and hodograph planes. For $0 < M < 1$ the standard method is applied in the stretched (Prandtl-Glauert) fcoordinates $(x, \beta y), (\beta u, v)$ where $\beta^2 = 1 - M^2$.

A. Implementation

The condition to be satisfied at a minimum (16) is a sum over the triangles surrounding a particular node. It can therefore be coded as a *distribution scheme*, in which the first step is to evaluate D, Ω in every triangle. The second step is to distribute the quantities to the nodes with appropriate weights. We can do this by duplicating a method of *steepest descent*,

$$\mathbf{U}_j^{n+1} = \mathbf{U}_j^n - \omega \nabla \mathcal{F}_j.$$

The operation within each triangle is to make the following changes at each node

$$\delta u_j = \omega[(1-M^2)D\Delta_j y + k^2\Omega\Delta_j x] \quad (22)$$
$$\delta v_j = \omega[-D\Delta_j x + k^2\Omega\Delta_j y] \quad (23)$$
$$\delta x_j = \omega[D\Delta_j v - k^2\Omega\Delta_j u] \quad (24)$$
$$\delta y_j = \omega[-(1-M^2)D\Delta_j u - k^2\Omega\Delta_j v] \quad (25)$$

where $\Delta_j(\cdot) = (\cdot)_{j-1} - (\cdot)_{j+1}$ is a difference taken along the opposite side of the triangle. The changes made to each node clearly sum to zero, so that in this sense the method is conservative. Note that with the particular choice of norm implied by $k = 1 - M^2$ some of these terms will be cancelled by contributions from other triangles and so need never be implemented. However, it is instructive to retain the full form for discussion. Note also the coding device that in any element where D or Ω is less than some tolerance one should jump out of the loop before doing any updating.

It is instructive to consider the way that the terms operate in pairs, for example the particular pair

$$\delta u_j = \omega[(1-M^2)D\Delta y_j], \quad (26)$$
$$\delta y_j = \omega[-(1-M^2)D\Delta u_j]. \quad (27)$$

These terms can be visualised as causing a certain kind of motion of the triangle in the plane u, y (Fig 3). Each point of the triangles is moved on a trajectory directed along the normal to the opposite side. Such a motion changes the area of the triangle in this particular projection as efficiently as possible. Cells for which D is large try to reduce their areas, and cells with small D do not mind increasing their areas. At a particular node there will be a competition in which each triangle tries to achieve its goals. This is a form of mesh adaptivity. Note that D also 'drives' a similar motion in the plane v, x, and that the algorithm overall is trying to bring about a constant of proportionality $(1 - M^2)$ between these two projected areas. In fact, if $D > 0$, the scheme tries to decrease the area in the u, y plane whilst increasing the area in the v, x plane, and *vice versa*.

The steepest descent method proves very slow in practice, and it is inherently poorly scaled. If the units of measurement are changed so that numerical values of u are doubled we would have to find that all changes to u double also. In fact it is the changes to y that are doubled. These problems

are solved by upgrading to a Newton strategy,

$$\mathbf{U}_j^{n+1} = \mathbf{U}_j^n - \omega \left[H^{-1}\nabla\mathcal{F}\right]_j .$$

where H is the Hessian of $\mathcal{F}$ with respect to $\mathbf{U}$.

To evaluate the Hessian, we write it in block form, with H_{ij} denoting the derivatives with respect to the variables stored at nodes i, j and note that $H_{ij} = 0$ unless either $i = j$ or else the nodes i, j are connected along an edge. From the definition of F, we have in the first case

$$H_{jj} = \sum_{T\in T_j} \left(\frac{\partial D}{\partial \mathbf{U}_j}\right)^2 + k^2 \left(\frac{\partial \Omega}{\partial \mathbf{U}_j}\right)^2 \qquad (28)$$

where the sum is over all triangles sharing the node j. In the second case,

$$H_{ij} = \sum_{T\in T_{ij}} \left(\frac{\partial D}{\partial \mathbf{U}_i}\right)^T \cdot \left(\frac{\partial D}{\partial \mathbf{U}_j}\right) + k^2 \left(\frac{\partial \Omega}{\partial \mathbf{U}_i}\right)^T \cdot \left(\frac{\partial \Omega}{\partial \mathbf{U}_j}\right) \qquad (29)$$

where the sum is over the two triangles that share the edge ij. Those terms that influence the updating of u, v reflect the geometry of the physical grid, and those terms that influence the updating of x, y reflect the 'geometry' of the hodograph grid.

A possibility is to retain only the diagonal terms in the Hessian. This removes the dimensional inconsistency from the steepest descent algorithm and creates something like a point Jacobi method that should be a good starting point for multigrid. There will, of course, be no change in the final converged solution on account of including either the full or 'diagonalised' Hessian.

Note however that components of the Hessian can vanish under certain conditions. These make sense. If all neighbours of j are at the same state (u, v) it no longer matters where node j is placed. Not quite so obviously, if all neighbours of j are at the same position (x, y) it no longer matters what values are assigned at j. In either case all residuals surrounding j vanish anyway. By following the above-mentioned strategy of jumping out of the loop whenever the residuals vanish these indeterminate cases, should they ever arise, will be left alone until such time as one of the nodes around j receives a perturbation from some other part of the grid.

B. Diagonal-Swapping

The method described so far preserves the mesh topology under the minimisation process. Clearly it is also possible to reduce $\mathcal{F}$ by comparing the sum of the residuals in any pair of triangles having a common edge (and therefore forming a quadrilateral) with the sum of residuals found for the triangles formed by drawing the other diagonal.

IV. The Supersonic Case

We will attempt a diagonalisation of the discrete equations in a slightly unusual sense. Consider the equations (22,23) describing the adjustments to u, v within a particular triangle. The adjustment to a particular linear combination $u + \lambda v$, say, can be written (with $\omega = 1$ for convenience; the actual value is irrelevant)

$$\begin{aligned}
\delta(u_j + \lambda v_j) &= [(1-M^2)D + \lambda k^2\Omega]\Delta_j y \\
&\quad -[\lambda D - k^2\Omega]\Delta_j x \\
&= \left[(1-M^2)^2 P\mathbf{uy} - (1-M^2)P\mathbf{vx}\right. \\
&\quad \left. +\lambda k^2 P\mathbf{vy} + \lambda k^2 P\mathbf{ux}\right]\Delta_j y \\
&\quad -\left[\lambda(1-M^2)P\mathbf{uy} - \lambda P\mathbf{vx}\right. \\
&\quad \left. -k^2 P\mathbf{vy} - k^2 P\mathbf{ux}\right]\Delta_j x \\
&= \left[P\{(1-M^2)^2\mathbf{u} + \lambda k^2\mathbf{v}\}\mathbf{y}\right. \\
&\quad \left. +P\{\lambda k^2\mathbf{u} - (1-M^2)\mathbf{v}\}\mathbf{x}\right]\Delta_j y \\
&\quad -\left[P\{\lambda(1-M^2)\mathbf{u} - k^2\mathbf{v}\}\mathbf{y}\right. \\
&\quad \left. -P\{k^2\mathbf{u} + \lambda\mathbf{v}\}\mathbf{x}\right]\Delta_j x
\end{aligned}$$

We will require that each term in braces in this last expression is proportional to the same linear combination of u, v, although not neccessarily the one appearing on the left. This leads to

$$\frac{(1-M^2)^2}{\lambda k^2} = -\frac{\lambda k^2}{1-M^2} = -\frac{\lambda(1-M^2)}{k^2} = \frac{k^2}{\lambda} .$$

For $M < 1$ these equations have no real solution, but for supersonic flow we have

$$k^2 = M^2 - 1, \qquad \lambda = \pm\sqrt{M^2-1} \qquad (30)$$

and with these choices we obtain a characteristic decoupling of the least-squares algorithm

$$\begin{aligned}
\delta(u_j + \lambda v_j) &= \lambda P(\lambda\mathbf{u} + \mathbf{v})(\lambda\mathbf{y} + \mathbf{x})\Delta_j(\lambda y + x) \qquad (31) \\
&= -\lambda(D - \lambda\Omega)\Delta_j(\lambda y + x) \qquad (32)
\end{aligned}$$

together with the same equation but the opposite choice of sign for λ.

This equation is driven, on the right, by differences of the characteristic variables in the physical $(\lambda u + v)$ and hodograph $(\lambda y + x)$ planes. A little surprisingly, the response on the left takes place normal

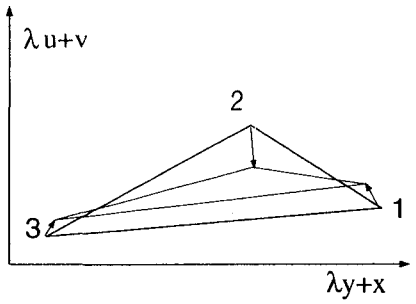
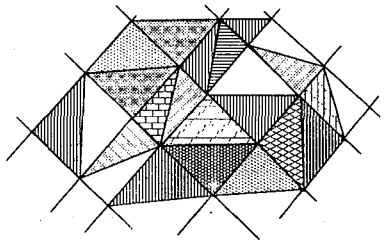


Figure 4: Node movement in the characteristic plane.



On the above grid, the supersonic system can be solved exactly, because it is possible for all of the residuals to be driven to zero precisely

Figure 5: A characteristic grid on which all cel residuals can be driven to zero.

to the characteristic direction. There are a similar pair of equations for the grid motion

$$\begin{aligned} \delta(y_j + \lambda x_j) &= \lambda P(\lambda \mathbf{y} + \mathbf{x})(\lambda \mathbf{u} + \mathbf{v})\Delta_j(\lambda u + v) \quad (33) \\ &= \lambda(D - \lambda\Omega)\Delta_j(\lambda u + v) \quad (34) \end{aligned}$$

Accepting the above choice of k^2 means that the particular norm for the cell error

$$F = \frac{1}{2}D^2 + \frac{|1 - M^2|}{2}\Omega^2 \quad (35)$$

has very distinguished properties in both the subsonic and supersonic cases.

It is interesting to observe the action of the scheme for a simple wave solution

$$\lambda u + v = \mathrm{Fn}\,(x + \lambda y)$$

The solution surface in 4-space collapses to a single curve in the projection corresponding to the characteristic variables $\lambda u + v, x + \lambda y$. Motion of the element nodes takes place in such a way as to reduce the area of each cell in this projection as quickly as possible, and will cease when all of the areas vanish.

Now a triangulated surface can only appear as a single (piecewise linear) curve when all the nodes are aligned along the characteristics. Therefore, when the minimum of $\mathcal{F}$ is achieved, that is $\mathcal{F} = 0$, the nodes will be so aligned and the solution is exact. It is also very nonunique; any set of nodes aligned in such a way can support an exact solution.

It is also possible to generate exact solutions for non-simple waves. Suppose that an element has one of its sides oriented along each characteristic. The orientation of the third side does not matter. If the corresponding characteristic equation is satisfied along each of the characteristic sides then it is easy to show that $F = 0$ for that element. Consider then a region covered by the characteristics stemming from some initial-boundary curve. If the grid points are fixed in this boundary, then a unique set of grid points is defined by the characteristics passing through them. The parallelograms formed by these may be divided by arbitrary diagonals to form a grid on which the exact solution may be found(see Fig ??). For such a solution $\mathcal{F} = 0$ and therefore the solution will be found by any procedure that minimises $\mathcal{F}$.

V. Three Dimensions

The Cauchy-Riemann system, viewed as a statement about analytic functions, has no meaning in three dimensions, but the small-perturbation flow equations, written as one divergence condition and two vorticity conditions, are very pertinent to many applications.

$$\begin{aligned} D = (1 - M^2)\partial_x u + \partial_y v + \partial_z w &= 0, \quad (36) \\ \Omega_z = \partial_x v - \partial_y u &= 0, \quad (37) \\ \Omega_y = \partial_x w - \partial_z u &= 0, \quad (38) \end{aligned}$$

Although the hodograph transformation is not normally regarded as relevant to three-dimensional flow either, there is no trouble extending the geometrical concepts. We consider the solution of the flow equations to be represented by a three-dimensional subspace embedded in the 6-space $(u, v, w, x, y, z)^T$. The discretisation of this subspace consists of tetrahedra.

Introduce the vector $\mathbf{x} = (x_1, x_2, x_3, x_4)^T$ with $\mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{v}, \mathbf{w}$ defined similarly. Let P be the third-rank tensor that computes the triple vector product, so that the geometrical volume of a tetrahedron is

$$V = \frac{1}{6}P\,\mathbf{xyz} \quad (39)$$

and recall that the sign of the volume will change with odd permutations of the vectors. Then we have,

within any small tetrahedron,

$$\partial_x u = \frac{P\,\mathbf{uyz}}{P\,\mathbf{xyz}}$$

because this expression is clearly exact whenever u is some linear combination of x, y, z. All other derivatives have analogous representations. A statement of the flow equations in terms of the geometry of 6-space is

$$F = \frac{1}{2}D^2 + \frac{|1-M^2|}{2}\left\{\Omega_y^2 + \Omega_z^2\right\} = 0 \qquad (40)$$

where the choice of norm is suggested by the two-dimensional analysis, and

$$\begin{aligned} D &= (1-M^2)P\,\mathbf{uyz} + P\,\mathbf{xvz} + P\,\mathbf{xyw} \\ \Omega_y &= P\,\mathbf{vyz} - P\,\mathbf{xuz} \\ \Omega_z &= P\,\mathbf{wyz} - P\,\mathbf{xyu} \end{aligned}$$

A numerical solution can proceed by minimizing the sum of F over all elements.

VI. Preliminary Results

At the time of preparing this paper, numerical results are only available for the special case of scalar advection,

$$a\partial_x u + b\partial_y u = 0. \qquad (41)$$

The residual of an element in $\mathbb{R}^3 = \{x, y, u\}$ can be written

$$R = aP\mathbf{uy} - bP\mathbf{ux}, \qquad (42)$$

and its first variation as

$$dR = P\mathbf{u}(a\mathbf{dy} - b\mathbf{dx}) - P(a\mathbf{y} - b\mathbf{x})\mathbf{du}. \qquad (43)$$

An algorithm that minimizes

$$\sum_T R_T^2$$

moves nodes normal to the characteristic directions until one edge in every cell is aligned with (a, b). However, such movement only takes place for those cells that experience a non-zero residual.

Results are given for a commonly-used test problem, circular advection in which $(a, b) = (-x, y)$. Input data along the negative x-axis should be output, mirror-imaged along the positive y-axis. Fig 6 shows an initial grid for this computation, comprised of square cells divided into triangles. To avoid the need for diagonal-swapping in this exploratory code, the sense of the initial diagonals is chosen favourably.

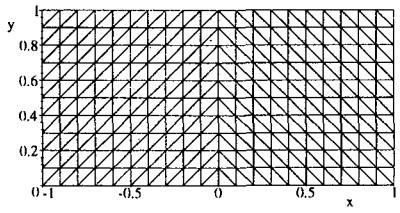


Figure 6: The initial grid for the circular advection problem.

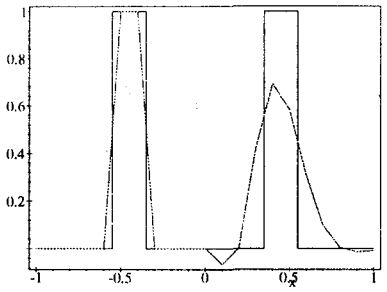


Figure 7: Results along the line $y = 0$ for the minimum residual algorithm on the fixed grid shown above.

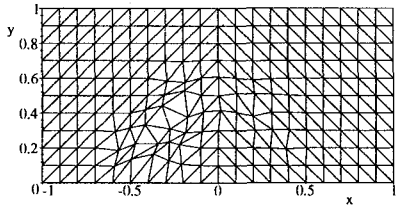


Figure 8: The grid begins to adapt as the solution develops.

Fig 7 shows the results obtained by minimising the residuals on this fixed grid.

When nodal movement is allowed, that part of the grid reached by the disturbance begins to move, as shown in Figure 8

Finally, at convergence, all of those cells that have been 'touched' by the developing solution have achieved alignment with the characteristics, and there are are two lines of edges whose end-points lie on two precise circles (Figs 9,10). The numerical solution is, to within machine error, exact.

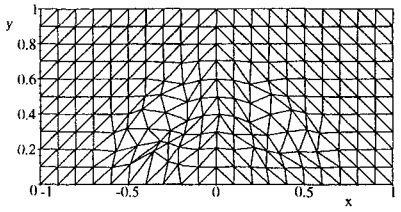


Figure 9: The final grid for the circular advection problem.

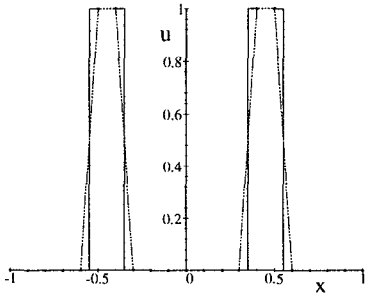


Figure 10: Results along the line $y = 0$ for the minimum residual algorithm on the final adapted grid.

VII. Summary and Conclusions

We have begun to lay foundations for a method that develops the solution and the grid in a unified manner. The duality between physical and hodograph representations of the Cauchy-Riemann equations provided initial inspiration, but the method does not seem restricted to such cases. If a problem involves m unknowns in d dimensions, it is reducible to studying the geometry of a d-dimensional manifold in an $(m + d)$-dimensional space..[9] 'Triangulating' this manifold, and minimizing some measure of the discrete error in each element leads automatically to a solution procedure that unifies the solution and the grid. Although not treated in this paper, boundary conditions appear naturally as a constraint on the minimization.

Future work will concentrate on two main issues. The norm of the discrete error is the key to success for large systems of equations, and the simple descent algorithm used here to generate the solutions is painfully slow and must be improved.

References

[1]d'Azevedo, E. F., and Simpson, R. B. (1991) On optimal triangular meshes for minimizing the gradient error, *Numerische Mathematik*, **59**, pp321-348.

[2]Baines, M. J. (1994) *Moving Finite Elements*, Oxford University Press.

[3]Dyn, N., Levin, D. and Rippa, S. (1990 Data-dependent triangulations for piecewise linear interpolation, *IMA J. Num. Anal.*, **10**, pp 137-154.

[4]Mesaros, L. and Roe, P. L. (1995) Multidimensional fluctuation-splitting schemes based on decomposition methods, *AIAA 12th CFD Conference*, pp 582-591.

[5]Paillère, H., Deconinck, H. and Roe, P. L. (1995) Conservative upwind residual-distribution schemes based on the steady characteristics of the Euler equations. *AIAA 12th CFD Conference,* pp 592-605.

[6]Rippa, S. (1992) Long and thin triangles can be good for linear interpolation, *SINUM*, **29** (1), pp 257-270.

[7]Roe, P. L., Compounded of many simples; the role of model problems in CFD, Proceedings of Workshop on Barriers and Challenges in CFD, NASALarc/ICASE, 1996, Kluwer to appear.

[8]Roe, P. L. and Mesaros, L. (1996) Solving steady mixed conservation laws by elliptic/hyperbolic splitting, 15th Int. Conf. Num. Meth. Fluid Dyn. *Lecture Notes in Physics*, Springer, to appear.

[9]Schutz, B., *Geometrical Methods of Mathematical Physics*, Cambridge, 1980.

[10]Tourigny,Y. and Hülsemann, F. (1997) A new moving mesh algorithm for the finite element solution of variational problems, *Math. Comp.*, submitted.