THE UNIVERSITY OF READING

The Use of Moving Grids in Contour Zoning

by

K.W. Blake

Numerical Analysis Report 8/99

DEPARTMENT OF MATHEMATICS

The Use of Moving Grids in Contour Zoning

K.W.Blake
Department of Mathematics
University of Reading
P O Box 220, Reading, RG6 6AX, UK

August 1999

Abstract

A moving grid method for the solution of partial differential equations by a contour zoning method is investigated. Starting on an unstructured triangular computational grid based upon discretised contours of an initial state, these contours and hence the grid are moved by preserving a simple equidistribution property.

In the Contour Zoning approach computational time is saved via a reduction in the number of equations to solve for the heights of the contours. In addition to this, the equidistribution is computed exactly and efficiently and not only moves the grid but positions nodes advantageously for greater efficiency and accuracy.

Applications are shown to insect dispersal models [3] involving non-linear diffusion equations in 1 and 2 dimensions.

1 Introduction

The contour zoning method involves a reduction in the number equations to be solved compared to more conventional numerical pde solvers, is made by having one equation for a set of similar valued grid points [1]. These sets can either be possibly, non-contiguous groups of points pre-determined from an initial estimate in steady state problems, or from a previous time-step in a non-steady state case. However the work presented here is based upon an unstructured triangular computational grid derived from contours of an initial state. When using such a grid, each equation now yields a value or 'height' for a particular contour. Sections 2 and 4 briefly outline the derivation of the contour zoning equations and the generation of the computational grid mentioned previously. For a more detailed and complete derivation of the resulting equations see [2].

An equidistribution principle outlined in section 4, and in [2] incorporated only in the construction of the triangular grid, is used to shift nodes and hence contours to more advantageous positions within the grid after each solution update. Despite the equidistribution involving the simplest of monitor functions, the ease of implementing this idea complements the reduction in the size of the system to solve for the values of the contours involved. Technically the algorithm involves the location and interpolation over edges connecting adjacent contours, but further savings are achieved by retaining connectivities between grid points throughout. This moving grid process is introduced in $1D$ and then naturally extends to higher dimensions.

Section 5 presents some numerical results in one and two dimensional versions of a non-linear diffusion problem. These results are compared to some analytic solutions relating to insect dispersal models found in [3]. Also included is a short note outlining a possible failure of the grid movement process where in severe problems, a permanent discontinuity is formed from which the solution never recovers.

Finally section 6 gives some final conclusions and some idea of possible future work including extensions to more complex monitor functions used in the equidistribution preservation.

2 Contour Based Grid

As mentioned earlier, all the two-dimensional work presented takes place on unstructured triangular computational grids constructed using discretized contours of the function u. An initial grid is generated from the contours of the initial state of the problem in hand.

These initial grids are constructed using MATLAB in two stages. Firstly, the nodes are found, which are quite simply points lying on an initial contour, the heights being chosen in accordance with the monitor function used in the equidistribution process. Equal numbers of nodes are placed on each contour in order to retain a reasonable resolution as the solution and hence the grid evolves.

Secondly, a triangulation is constructed between each adjacent discretised contour. Currently the type of triangulation used is thought to be of little significance, since the solution values of the connected nodes can only take one of the heights of the contours in question and hence the spatial derivatives between the contours are somewhat limited. The main triangulation used here is by Delaunay (see [5]) although it may be necessary to remove any triangles connecting nodes belonging to the same contour. These are not required, since spatial gradients along contours always equal zero.

Fig 1. shows an example of such a grid. Here the grid is generated for a radially symmetric piecewise continuous initial condition typical of the family of solutions found in insect dispersal problems [3] which are later used in section 5. In this relatively simple example, 21 contours are discretized (including the maximum at $(0,0)$), and each contour contains 15 nodes. The underlying monitor function is taken as $M(u) = \frac{\mathrm{d}u}{\nabla u}$, resulting in the heights of the contours being equally spaced between the minimum and maximum of u as shown in the left hand graph.

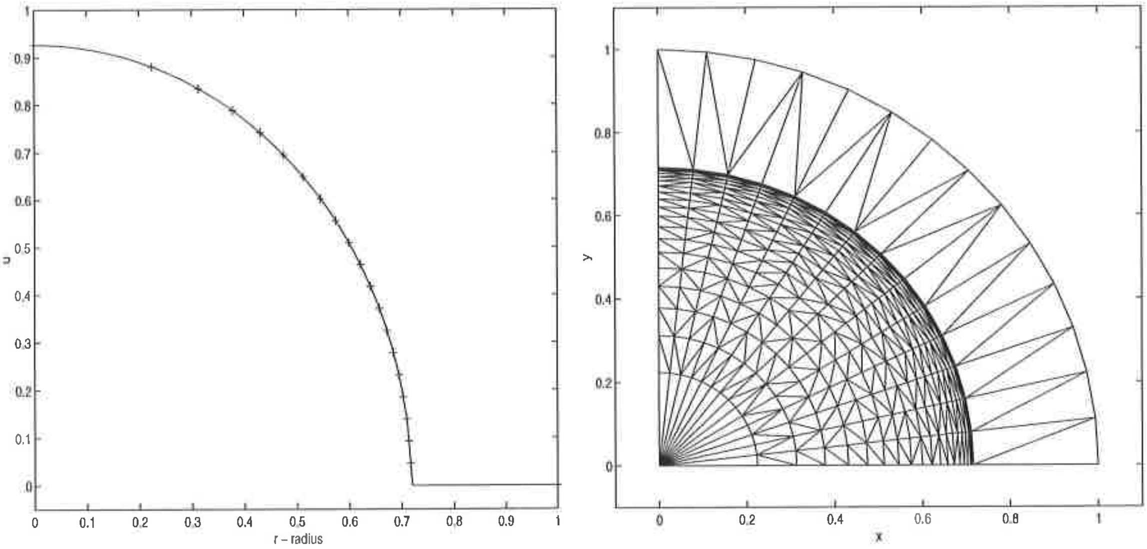

Figure 1: Example of Contour Based Grid with associated radial cross-section

3 The Contour Zoning Equations

To demonstrate the derivation of the contour zoning equations we consider a non- linear diffusion problem

$$\frac{\partial u}{\partial t} = \nabla.(u^m \nabla u) \tag{1}$$

where m is real.

The equations are formed by considering the integral form of (1) over the iz^{th} contour (2), applying Gauss' theorem and then summing discretized, semi-implicit fluxes out of some predefined control volume (see below) relating to the iz^{th} contour. Thus we work with the form

$$\begin{aligned} V_{iz}\frac{\partial u}{\partial t} &= \int\int_{V_{iz}} \nabla.(u^m \nabla u) dxdy \\ &= \int_{S_i} u^m \nabla u.dS \end{aligned}$$

Where S_i is the boundary of the control volume , described below.

Let us consider an interior node i in the triangular mesh. As in [4], we define a secondary mesh of irregular polygons whose vertices are alternately the centres and the midpoints of the sides of the adjacent triangles to which node i belongs, see Fig.2. Collecting together the secondary mesh elements of nodes lying on the i^{th} contour gives the control volume of the contour, which has area V_{iz}. The right-hand side of equation (2) is equal to the integral around the boundary of V_{iz} of the normal flux of the diffusing quantity u, which is in turn equivalent to summing fluxes out of each discretized section of V_{iz} into adjacent control volumes. A conventional semi-implicit time discretization deals with the left hand side of equation 2.

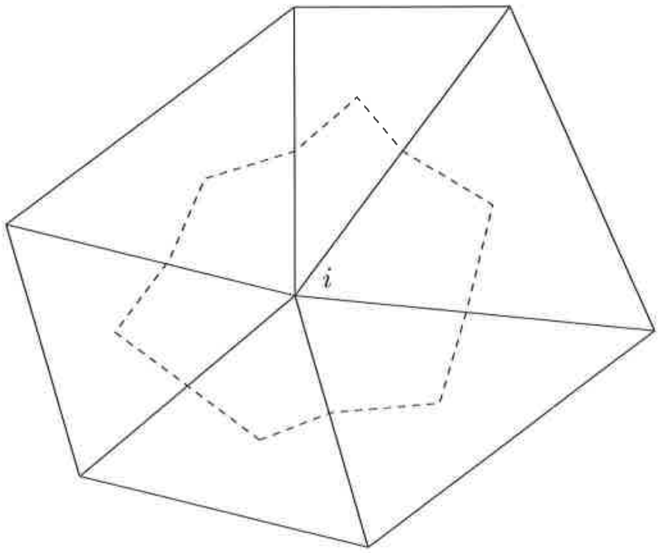

Figure 2: Secondary mesh element associated with an interior node

Fig. 3, shows node i with three neighbouring nodes k,j and m. The vectors **k**,**j** and **m** define the connecting edges of the primary edge and the vectors $\mathbf{b}_{ikj}$ and $\mathbf{b}_{ijm}$ define two sections of the boundary of V_{iz}. The flux of material out of section $\mathbf{b}_{ikj}$ is equal to the scalar product of some linear vector gradient function from $\triangle_{ikj}$ and the outward normal to $\mathbf{b}_{ikj}$. Similarly with section $\mathbf{b}_{ijm}$. The sum of these two contributions can be written as

$$C_k u_{kz}^{n+1} + C_j u_{jz}^{n+1} - C_m u_{mz}^{n+1} + (C_m - C_j - C_k) u_{iz}^{n+1}$$

where u_{kz}^{n+1},u_{jz}^{n+1} and u_{mz}^{n+1} are the implicit 'heights' of the contours to which nodes k,j and m belong. The co-efficients C_k,C_j and C_m can be written in terms of the vectors defined in Fig 3 as follows.

$$C_k = \begin{cases} (u_{ikj}^n)^m \dfrac{\widetilde{\mathbf{j}}.\widehat{\mathbf{b}}_{ikj}}{\mathbf{k}.\widetilde{\mathbf{j}}} & \text{where } u_{iz} \neq u_{kz} \\ 0 & \text{where } u_{iz} = u_{kz}. \end{cases}$$

$$C_j = \begin{cases} (u_{ijm}^n)^m \dfrac{\widetilde{\mathbf{m}}.\widehat{\mathbf{b}}_{ijm}}{\mathbf{j}.\widetilde{\mathbf{m}}} - (u_{ikj}^n)^m \dfrac{\widetilde{\mathbf{k}}.\widehat{\mathbf{b}}_{ikj}}{\mathbf{k}.\widetilde{\mathbf{j}}} & \text{where } u_{iz} \neq u_{jz} \\ 0 & \text{where } u_{iz} = u_{jz}. \end{cases}$$

$$C_m = \begin{cases} (u_{ijm}^n)^m \dfrac{\widetilde{\mathbf{j}}.\widehat{\mathbf{b}}_{ijm}}{\mathbf{j}.\widetilde{\mathbf{m}}} & \text{where } u_{iz} \neq u_{mz} \\ 0 & \text{where } u_{iz} = u_{mz}. \end{cases}$$

where the vector $\widetilde{\mathbf{j}}$ is vector $\mathbf{j}$ rotated clockwise by $\frac{\pi}{2}$, $\widehat{\mathbf{b}}_{ikj}$ is the outward normal of $\mathbf{b}_{ikj}$ and $(u_{ijm}^n)^m$ is the semi-implicit non-linear co-efficient evaluated at the centre of $\triangle_{ijm}$. For a more detailed derivation of these definitions and expressions see [2].

Collecting together the contributions from all section of V_{iz}, we now have an equation for u evaluated on the i^{th} contour.The limits on the first summation denote all the nodes i belonging to contour iz, while the second summation denotes all the neighbouring nodes j of i. When re-arranged, this leads to the $N \times N$ tri-diagonal system 2. Boundary conditions are easily imposed on the relevant parts of the affected contours.

$$V_{iz} \frac{u_{iz}^{n+1} - u_{iz}^{n}}{dt} = \sum_{\forall i \in iz} \sum_{j} C_k u_{kz}^{n+1} + C_j u_{jz}^{n+1} - C_m u_{mz}^{n+1} + (C_m - C_j - C_k) u_{iz}^{n+1} \quad (2)$$

4 Moving Contours

The movement of contours and hence the grid is motivated by the preservation of an easily enforced equidistribution principle. After each time step the nodes are moved such that they are equidistributed, that is they satisfy

$$\int_{x_i}^{x_{i+1}} M(u) = constant \quad \forall i = 1, N \quad (3)$$

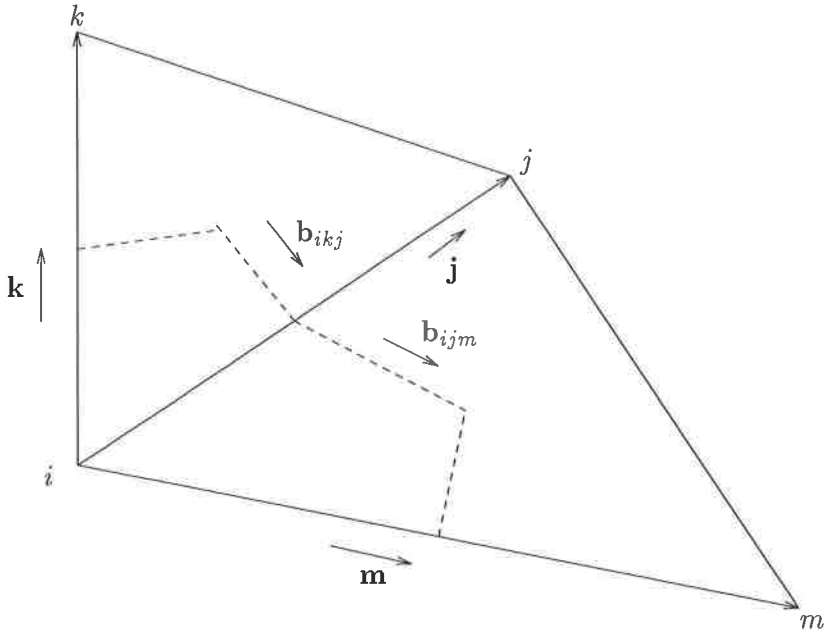

Figure 3: Nodes i,j and associated vectors.

whre $M(u)$ is some Monitor function

Now let us take $M(u)$ such that

$$M(u) = |\nabla u|.$$

It can be shown that for a monotonically decreasing function u, the values of u at the newly equidistributed points will be equally spaced between the minimum and maximum values of u, so that for N nodes,

$$u_i = u_{max} - (i-1)\frac{(u_{max} - u_{min})}{N-1} \quad \forall i, i = 1, N$$

To illustrate the algorithm, Fig. 4 shows the three main stages in a complete time step. Starting with a freshly equidistributed set of nodes and values in Fig. 4 (i), the diffusion equation (1) is solved for the next time level using the contour zoning equations (2), the new values of u being shown in Fig. 4 (ii), whre no node movement has taken place yet. From this new solution profile, the new equidistributed contour 'heights' are computed from (4), then via interpolation over the profile shown in Fig. 4 (ii) the positions of the new nodes can be found. The newly equidistributed solution is shown in Fig. 4 (iii) ready for the next computation.

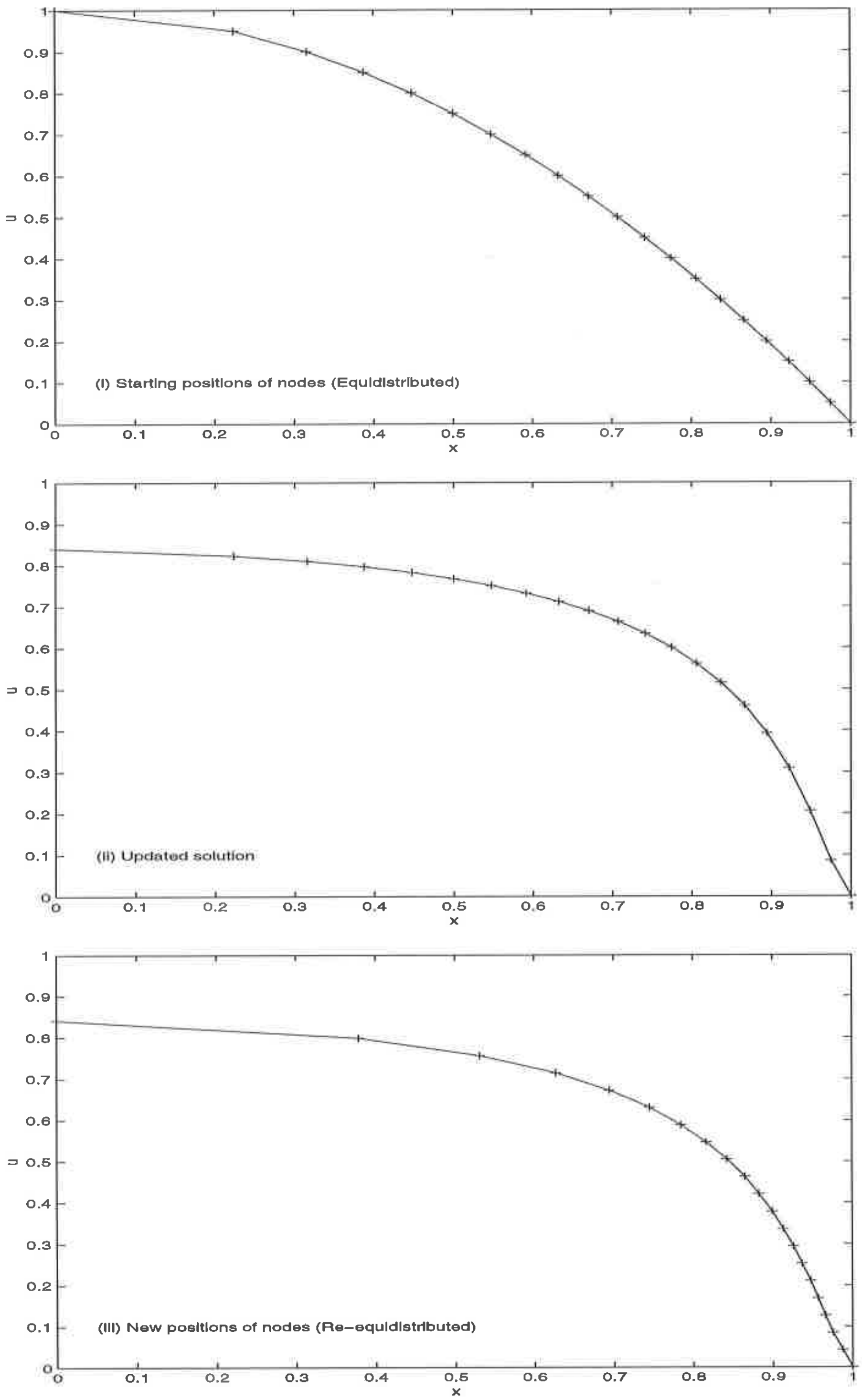

Figure 4: The three stages of node movement

In two dimensions, the algorithm is unchanged, although an interpolation process is now needed to find a set of nodes to represent a contour, rather than to locate a single point. Fig.5 shows two contours i and j with newly computed values, or 'heights' u_{iz}^{n+1} and u_{jz}^{n+1}. Now let us suppose that a newly equidistributed contour level u_m^{n+1} lies between these two heights. Linear interpolation takes place along all edges of the grid connecting nodes belonging to contours i and j. We now have a discretization of the newly placed contour denoted by the broken line. However it is noticed that the new contour is defined at nearly twice as many nodes as contours i and j. So that the new contours has the correct number of nodes such that connectivities throughout the mesh remain constant, we have to interpolate again, this time along u_m^{n+1}, which is done in such a way that nodes are again equally spaced over the contour.

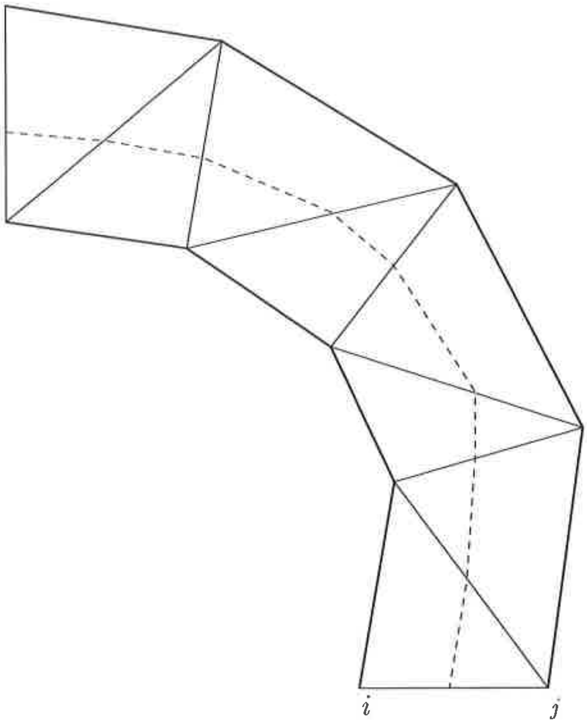

Figure 5: Defining a new contour in Two-Dimensions

5 Numerical Results

Taking advantage of reference solutions in one and two dimensions from [3], we are able to asses our approximated solutions to equation (1). The test problems are based on insect dispersal models. The contour zoning solutions take an initial state from one of these solutions at an early arbitrary time $tstart$.

5.1 One-Dimension

In one dimension equation 1 reduces to

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}(u^m \frac{\partial u}{\partial x})$$

and has solution

$$u(x,t) = \begin{cases} \frac{1}{\lambda(t)}[1 - \{\frac{x}{r_0\lambda(t)}\}^2]^{\frac{1}{m}} & |x| \leq r_0\lambda(t) \\ 0 & |x| > r_0\lambda(t). \end{cases}$$

where $\lambda(t) = \left(\frac{t}{t_0}\right)^{\frac{1}{(2+m)}}$, $r_0 = \frac{Q\Gamma(\frac{1}{m}+\frac{3}{2})}{\pi^{\frac{1}{2}}\Gamma(\frac{1}{m}+1)}$ and $t_0 = \frac{r_0^2 m}{2(m+2)}$, Γ is the gamma function and Q is the quantity of material,(or insects) released initially at the origin.

The solution (5.1), represents a kind of wave with the front at $x = x_f = r_0\lambda(t)$. The derivative of u is discontinuous here. Neumann boundary conditions are imposed at $x = 0$ and also at the foot of the front. The wave 'front' which is defined as the point where $u = 0$, propagates very quickly initially, and slows down with time until $\frac{\partial u}{\partial t} \rightarrow 0$ as $t \rightarrow \infty$.

Since the conservation of the amount Q drives the solution (5.1), Neumann boundary conditions are implemented at $x = 0$ and at a fixed contour at $x = 1$. The contour zoning solution is then generated with $Q = 1$,$m = 3$ and $tstart = 0.01$. Also included in the contour zoning algorithm is a variable time step dt. The length of time step is chosen so that the solution will decay at its maximum by a pre-determined constant amount, dc. The step length is computed by taking a implicit version of the contour zoning equations (2) at the maximum value of u, and then substituting in $dc = u^{n+1}_{maxz} - u^n_{maxz}$. Finally a maximum is imposed on dt such that $dt \leq dtmax = 0.01$ to ensure that too large a time step is not taken as the movement of the front slows down. So we have

$$dt = \frac{dcV_{maxz}}{|\Psi_{max}|}$$

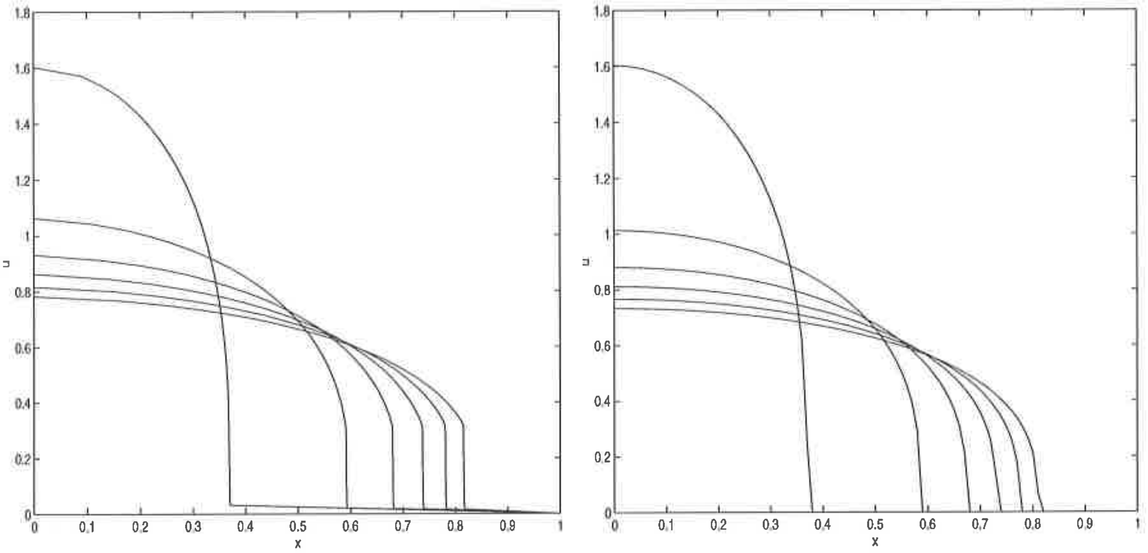

Figure 6: Approximate Solution (Left) and Reference Solution (Right) at times $t = 0, 0.1, 0.2, 0.3, 0.4, 0.5$

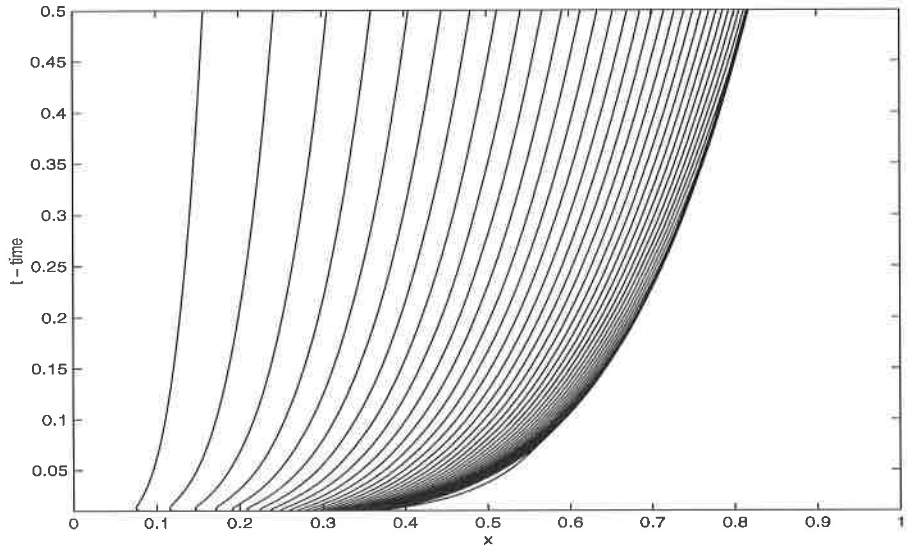

Figure 7: Trajectories of Nodes

where $\Psi_{max} = \sum_{\forall i \in iz} \sum_j C_k u^n_{kz} + C_j u^n_{jz} - C_m u^n_{mz} + (C_m - C_j - C_k) u^n_{maxz}$

The left-hand side graph in Fig. 6 shows the contour zoning solution for various times, whilst the right-hand side shows the associated reference solutions according to [3]. Generally the solution seems to be good, although there are small flaws. The solution has almost formed a 'shallow ramp' at the front and maybe because of this and the conservation of mass, the height of the maximum of u at $x = 0$ differs slightly from the analytic solution. Fig. 7 shows the trajectories of the nodes with time, and it is clear that the nodes at the foot of the front have clustered and seemed to 'stick' together to form the steep front noticed in Fig. 6. These results, in one-dimension, only really demonstrate the effectiveness of the node movement and not the efficiency of the contour zoning formulation, since in $1D$ the equations become similar to a conventional finite-difference method and hence there is no reduction in the number of equations to solve.

5.2 Two-Dimensions

We now compare a two-dimensional contour zoning solution to the corresponding plane radially symmetric problem, which has solution

$$u(r,t) = \begin{cases} \frac{1}{\lambda(t)^2}[1 - \{\frac{r}{r_0\lambda(t)}\}^2]^{\frac{1}{m}} & r \leq r_0\lambda(t) \\ 0 & r > r_0\lambda(t). \end{cases}$$

where $\lambda(t) = \left(\frac{t}{t_0}\right)^{\frac{1}{2(1+m)}}$, $r_0^2 = \frac{Q}{\pi}(1 + \frac{1}{m})$ and $t_0 = \frac{r_0^2}{4(m+1)}$. Again Q is the number of insects or amount of material initially at $r = 0$.

Considering a quarter section of the whole domain, and using a maximum radius of $r = 1$ an initial grid is generated from the contours of the solution above at $tstart = 0.01$. Fig. 8, shows the grid along with a 3-D plot of the the corresponding initial state. The Neumann boundary conditions are imposed at the contour of maximum radius, so the generated solution will become invalid as the propagating front moves past this final contour. The grids contain 51 contours, each being discretised by 20 nodes (except for the maximum at $[0, 0]$). Setting the variable time step parameter dc to be equal to 0.01, $Q = 1$, and running the solution until $t = 0.5$, the solution completed 511 time steps (each including solution update and grid movement) at an average time of 0.0705 secs per time step on a SUN ULTRA 5 workstation.

To illustrate the grid movement and in particular how the contours have positioned themselves onto the front, the final grid and its 3D plot are shown in Fig. 9.

Fig. 10 shows cross-sections of the approximate and exact solutions at various times. The excessively steep front formation which was noticed in the one-dimensional approximate solution have not formed as extensively here, a possible explanation for which being that in this case the same quantity of insects/amount of material is released over a larger space and hence the fluxes involved are not as severe. What is noticeable is the error in the solution at the very bottom of the front, since the equidistribution process used will always produce this shallow 'ramp' where the solution in these areas should be flat. Apart from these discrepencies the solutions seem to match quite well.

To see the impact of the grid movement used with the adaptive time stepping, we can compare the solution to a solution computed on the same initial grid but with the grid staying stationary throughout. Although the contour zoning equations have been shown to give adequate results on less severe problems (see [2]), the advantages of a moving grid are clear when considering the problem with $m = 3$. Fig. (11) shows the cross-section of the solution generated with no grid movement, for comparison with the solution with grid movement included see the left hand side of Fig. 10. It is clear that the moving grid actually allows the solution to not only form but also to help move the propagating front outwards.

5.3 Permanent Front Formation

Although good results have been shown above, it has been noted, especially in the $1D$ work, that unless the adaptive time-stepping parameter dc is chosen to be small enough the moving contours can become stuck, resulting in a sharp discontinuity where all the moving contours are very densely packed into a very small interval based around or on the forming discontinuity. Fig 12 gives a brief explanation as to how the contours or nodes become stuck in this way.The left hand graph shows an initial state with equidistributed nodes, the middle graphs shows the updated solution at the current nodes while finally the right-hand side shows the re-equidistributed nodal positions of the updated state. As can be seen more and more of the nodes have been 'sucked' into the front at $x \approx 0.4$. If this process was continued then all nodes except those at $x = 0$ and $x = 1$ would be clustered very tightly around this position and a sharp discontinuity would form from which the solution never recovers. However it has also been noted that by choosing a very small value for dc this negative property can be avoided and overcome.

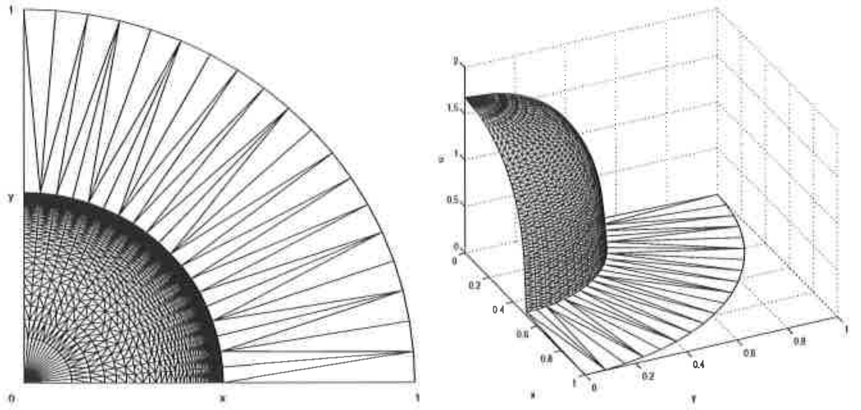

Figure 8: Initial Grid (Left) and its corresponding 3D plot (Right)

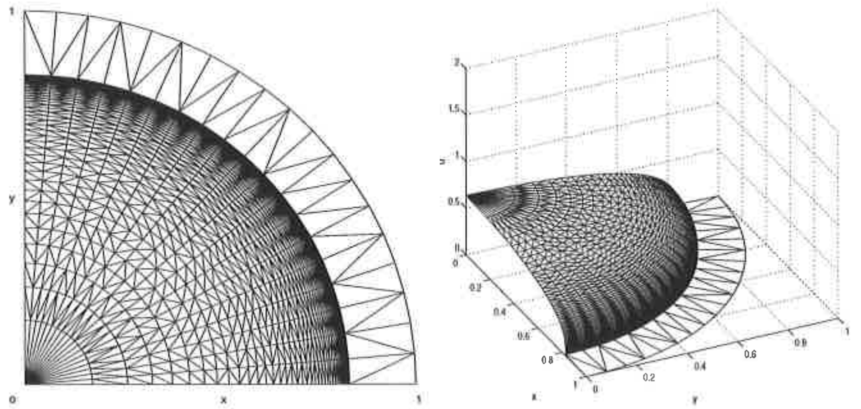

Figure 9: Final Grid (Left) and its corresponding 3D plot (Right)

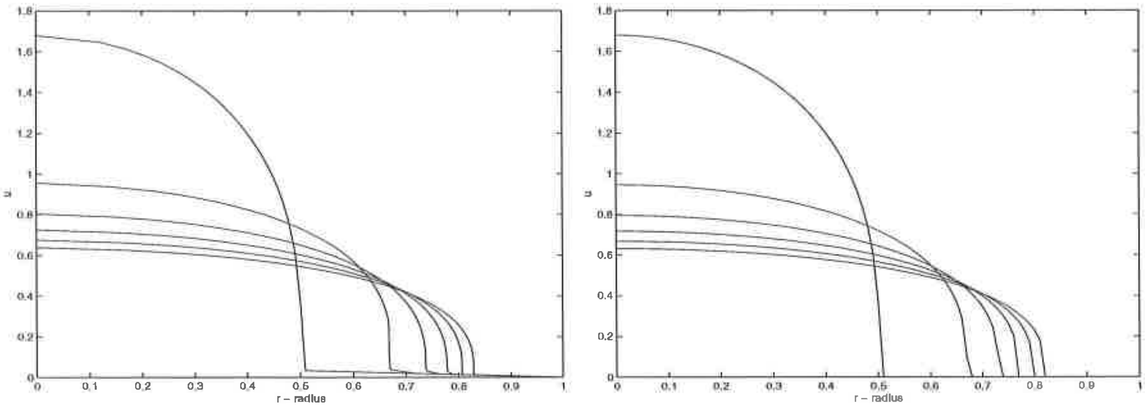

Figure 10: Cross sections of the approximate (Left) and the exact (Right) solutions at $t = 0, 0.1, 0.2, 0.3, 0.4, 0.5$

6 Conclusions and Further Work

It has been shown that the introduction of moving contours greatly improves the accuracy of the contour zoning method. The preservation of the equidistribution principle allows the solution to accurately resolve and track the moving fronts involved in the problems attempted. Moreover the computational effort involved when using such a simple monitor function compliments the reduction in work from the formulation of the contour zoning equations.

However small problems such as the shallow ramp (a data representation problem (see Fig 10)) and the 'permanent front' formation both seem to stem from the simple nature of the underlying monitor function. It can be shown that by using a more sophisticated function, these problem can be remedied and the solutions and hence method be enhanced further. As an example we consider the Modified Fisher Equation.

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}(u\frac{\partial u}{\partial x}) + u(1-u) \tag{4}$$

Fig 13 shows two sets of contour zoning results. On the left-hand side, the moving contour zoning method as before with the monitor function $M(u) = \frac{\partial u}{\partial x}$. Here it can be seen that the shallow ramp at the the foot of the initial state has been reacted upon and hence driven the foot of the front upwards so that the solution finally becomes fully reacted everywhere. The right-hand side of Fig. 13 however, shows results from the same problem but uses the arc-length monitor (see [6]). In this case there is now no 'shallow ramp' and hence the reaction only takes place behind the wave front as required. It is also noted that the results shown on the right-hand side of Fig 13 compare favourably with the numerical results presented in [7].

Other planned work for the future includes the inclusion of an independently moving 'last' contour. In the work presented above there was always a fixed 'last' or minimum valued contour at which the boundary conditions were implemented. This meant that when the front moved beyond this point or contour the solution became invalid. It is hoped that by moving this final contour with time the boundary condition may be imposed directly at the foot of the moving front and hence improve the solution further.

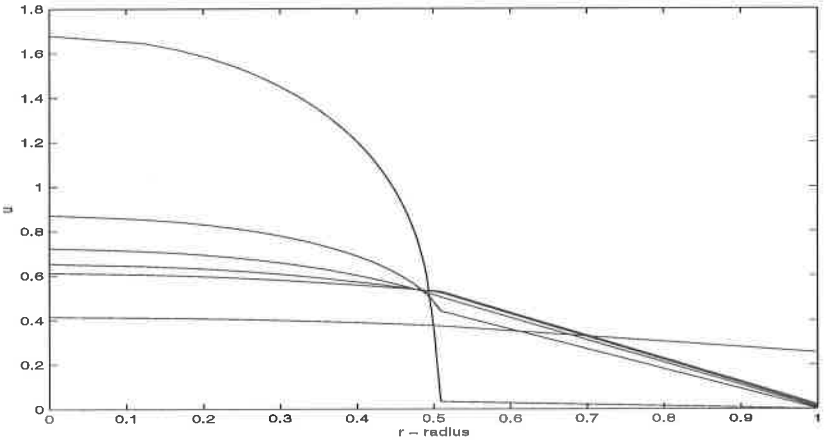

Figure 11: Cross section of approximate solution with $m = 3$ with no grid movement. Solutions are at $t = 0, 0.1, 0.2, 0.3, 0.4, 0.5$

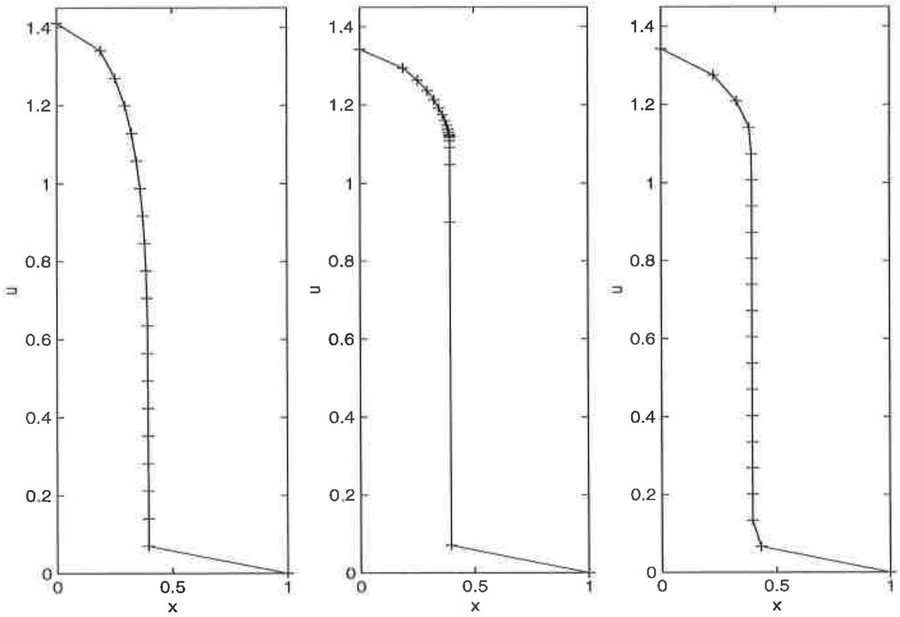

Figure 12: Formation of a 'permanent' front

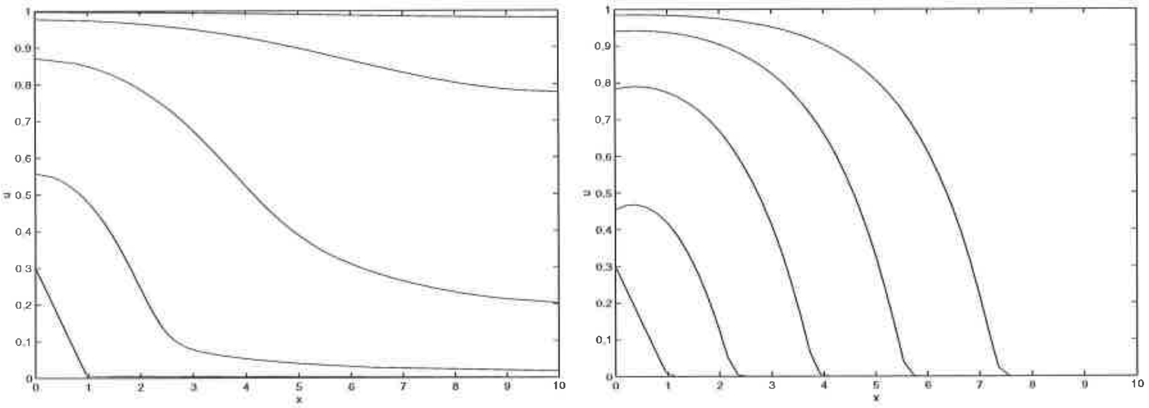

Figure 13: Moving Contour Zoning results to equation 4 for times $t = 0, 2.5, 5, 7.5, 10$

References

[1] Inston H.H., Contour Zoning, March 1998.

[2] Blake K.W, Baines M.J, New Developments in Contour Zoning, University of Reading, Department of Mathematics, Numerical Analysis Report 5/1999.

[3] Murray J.D, Mathematical Biology, Springer 1993

[4] Winslow A.M., Numerical Solution of the Quasilinear Poisson Equation in a Nonuniform Triangle Mesh, Journal of Computational Physics, Vol 1,p128-136, 1967.

[5] Malcolm A.J, Data Dependent Grid Generation, Ph.D. Thesis, University of Reading, Department of Mathematics, 12/91.

[6] Baines M.J, Grid Adaptation via node movement, Applied Numerical Mathematics 26, p77-96, 1998

[7] Needham D.J. and King A.C, The effects of variable diffusivity on the development of travelling waves in a class of reaction-diffusion equations, Royal Society of London 1994 348, p229-260,