

An Analysis on Moving Mesh Methods for Non-Linear Partial Differential Equations: the Approximate Solution

Sikander*

Assistant Professor of Mathematics, A.I.J.H.M. College Rohtak, Haryana, India

Abstract – This study is primarily concerned with the use of moving mesh methods for the approximate solution of non-linear partial differential equations of parabolic type. Such methods have become a popular means for the solution of problems which may contain sharp features that are hard to approximate. Whilst efficiently managing computational overheads. Initially, a novel moving grid technique known as Contour Zoning is discussed. This 'static' method is able to reduce numerical resources by grouping together sets of nodes as a moving contour of the solution.

-----X-----

INTRODUCTION

The use of adapted meshes in the numerical solution of partial differential equations (PDEs) has become a popular technique for improving existing approximation schemes. In problems in which features with large solution variations are common. Such as steep fronts and sharp variations, the choice of a non-uniform mesh can not only retain the accuracy but also improve the efficiency of an existing method by concentrating mesh points within regions of interest. This study is primarily concerned with the use of such methods for the solution of non-linear parabolic PDEs of the form

$$u_t = (D(u)u_x)_x + Q(u).$$

In particular, we shall be considering problems involving non-linear diffusion and solution blow-up.

The advantages of such an approach go hand in hand. Firstly, since such areas of interest in the mesh inevitably involve large variations in the solution, for any numerical scheme a smaller spatial resolution in the mesh is essential for a reliable approximation and accurate representation. However, to enforce this requirement over the entire grid will be an expensive process, especially in higher dimensions. It becomes obvious then, that for efficiency it is desirable to concentrate nodes and hence computational effort in those parts of the grid that require most attention. A successful approach will then ensure suitable mesh resolution whilst retaining computational efficiency.

In general there are three classifications of grid adaption. The first, h-refinement, adds extra nodes to an existing mesh to improve local grid resolution. A second technique, p-refinement, employs higher order numerical schemes to improve local accuracy as well as to approximate troublesome derivatives. The third approach is refinement, which maintains the existing number of nodes globally but relocates them strategically and, more importantly, efficiently over the domain, it is this latter idea that we shall be concerned with in the course of this study.

For static methods, the approximate solution is defined initially on a given mesh. During the calculation a new mesh, which may or may not have the same number of nodes, is generated using an existing grid generation technique. The solution is then interpolated onto the new mesh, so the redistribution and/or addition of grid points and the interpolation are all carried out at a fixed time in the solution process. Although successful, these methods carry large computational overheads, due to the intermittent changes in the data structure describing the mesh in any arithmetic code.

In dynamic methods, a mesh equation is employed to prescribe the speeds of nodes in order to move a mesh in such a way that gridpoints remain concentrated in regions of rapid variation as the solution evolves with time. In general in these methods the number of nodes in the grid remain the same. For this type of moving mesh method two coupled equations need to be considered, the moving mesh equation controlling the development of the mesh and that associated with the underlying problem. The mesh then develops continuously with no interpolation steps required. During this study we

shall look at both types of moving mesh method, with the principle interest lying in the former.

Both static and dynamic methods require some underlying motivation for their distributive strategies. It seems that in general the principles behind moving a grid through time employ the same techniques involved in generating a stationary mesh, for example a static method may redistribute grid points via an equidistribution principle. Indeed, many dynamic methods are derived by introducing a time derivative into an existing grid generation technique. For instance, in the functional minimisation approach to grid generation moving mesh equations can be derived from solving the gradient flow equations associated with the minimal grid functional, see Huang and Russell.

This study begins by outlining some of the existing literature on both grid generation and moving mesh methods. The next chapter will introduce both the equidis, tribution and functional minimisation approaches to grid generation. It is hoped that the reader, by first understanding the aims and means of grid generation, will find it easier to grasp the implementation of these techniques when applied to moving mesh problems. We will introduce some self, similar theory with specific reference to the porous medium equation. PME for which an analytical solution exists in. We shall take advantage of both these particular solutions and the selfsimilar qualities in later chapters to verify our numerical computations.

GRID GENERATION AND MOVING MESH METHODS

In general, moving mesh methods are derived from introducing node speeds, i.e. velocities of computational nodes, into existing algorithms for generating computationally advantageous meshes for steady state problems. An obvious example of this development is the several variations of moving mesh partial differential equations (MMPDES). Here a simple equidistribution relation in one spatial dimensional is differentiated with respect to time in order to derive equations prescribing the correct velocities of nodes in order to preserve the equidistribution principle as the solution and grid evolve. In higher dimensions due to the lack of a strict extension of the equidistribution idea in more than one spatial dimension, a popular idea is to evolve mesh speeds by attempting to keep a functional concerned with static grid generation minimal.

Grid Generation Methods and Techniques-

The simplest place to start an exposition of the basic philosophy behind the use of an adapted, irregular grid is in one dimension. The most widely used method is the equidistributed mesh. The principles of the method were later applied to generating efficient computational grids for the numerical solution of steady PDEs. For example White used a transformation to arc-length

coordinates to generate equidistributing meshes for the numerical solution of two-point boundary value problems. In another approach where the one-dimensional mesh was iterated by trying to reduce the truncation error of the solution of the underlying PDE after each iteration. This is a convenient point at which to formally introduce and dene the equidistribution principle.

The main strategy behind the equidistribution idea is quite self-explanatory. The idea is to choose a mesh such that a measure of either the geometry of the represented function, or of the error of the numerical solution, is distributed equally between adjacent nodes. This measure is prescribed via a user-defined function known as the monitor, a positive-definite function of the solution u and/or its derivatives u_x, u_{xx} , of the form.

$$M = M(x, u, u_x, u_{xx}). \quad (2.1)$$

Later on in this section, we shall introduce various choices of monitor function and illustrate their effect on the resulting mesh. However, we begin by stating how this measure is distributed over the grid in a formal definition.

Given a mesh representing a physical space in one-dimension $x \in [a, b]$ with $N+1$ mesh points $x_i, i = 0, \dots, N$, such that $x_0 = a$ and $x_N = b$, the equidistribution principle can be written

$$\int_{x_i}^{x_{i+1}} M dx = \frac{1}{N} \int_a^b M dx \quad i = 0, \dots, N-1. \quad (2.2)$$

However, in most grid generation applications it is often more convenient to think of the equidistribution idea as one of a coordinate mapping from a computational, space to a physical one. The goal of the grid generation problem then becomes one of finding a suitable coordinate mapping or transform. This approach is common and forms the basis of most grid generation techniques. and, indeed, moving mesh methods. Concentrating still on one dimension, we define the computational space $\xi \in [0, 1]$, so that the mesh points in physical space are related to the (usually regularly spaced) grid points ξ_i in the computational domain. Written formally, x is then a mapping from ξ to x

$$x = x(\xi).$$

Within this framework the equidistribution idea is written as

$$\int_0^{x(\xi_i)} M d\tilde{x} = \xi_i \int_0^1 M d\tilde{x} \quad (2.3)$$

or

$$\int_{x(\xi_i)}^{x(\xi_{i+1})} M d\tilde{x} = \frac{1}{N} \int_0^1 M d\tilde{x}. \quad (2.4)$$

Differentiating (2.3) with respect to ξ once gives the equation differentiating yet again yields the equation.

$$\frac{\partial}{\partial \xi} \left(M \frac{\partial x}{\partial \xi} \right) = 0. \quad (2.5)$$

Following this approach, the solution of (2.5) with Dirichlet boundary conditions

$$x(0) = a \quad x(1) = b$$

produces an equidistributed grid for the given monitor function. However, equation (2.5) is non-linear since M depends not only on x but also on the solution u . To overcome this, an iterative approach is suggested using the algorithm

$$(M(x^p)x_{\xi}^{p+1})_{\xi} = 0 \quad (p = 0, 1, \dots)$$

which may be discretised in a semi-implicit style as follows.

$$M(x_{i+\frac{1}{2}}^p)(x_{i+1}^{p+1} - x_i^{p+1}) - M(x_{i-\frac{1}{2}}^p)(x_i^{p+1} - x_{i-1}^{p+1}) = 0. \quad (2.6)$$

The resulting tridiagonal system is easily solved using, for example, a Jacobi-iteration method.

When generating an equidistributed grid for good representation of a function or initial condition, the values of the monitor are known exactly and the iteration is usually quick and successful. However when using this type of iterative process for adapting a mesh to give a better numerical solution to an underlying differential equation, it is common to use an interleaving approach where the grid and solution are alternately updated, with the solution being interpolated between changing states of the mesh.

Moving Mesh Methods -

In the previous section, we have outlined the aims and some techniques behind the generation of irregular grids. We now turn our attention to methods which aim to move the mesh in time to solve non-steady differential equations. whilst retaining the properties (and hence the numerical benefits) of the ideas presented above.

An early incorporation of the equidistribution idea into a moving mesh method is outlined by Petzold. Here a natural extension of the interleaving numerical solution

approach for a stationary, adaptive grid is presented. Since the solution of the problem now develops with time, the equidistribution part of the interleaving solution approach is undertaken at intervals, usually chosen by some predetermined error measure, during the forward integration in time. In other words, at certain times throughout the numerical solution of the equation, the grid is reequidistributed, hence moving the nodes throughout time, the solution on the new grid being found via some interpolation process. In a slight variation on this technique Blom et al used a predictive step, reequidistribute the grid using the prediction and then update the solution on the new grid. The update step is written in a Lagrangian form, involving the movement of the nodes in the redistribution, hence no interpolation step is required. The Blom approach bridges the gap between the static, regridding technique of Petzold and more dynamic traditional moving mesh methods. The major difference between the two is the interpretation of mesh speeds included within the solution procedure. We continue this theme further and explore the various forms of this continuously moving mesh idea.

We now turn our attention to the work of Huang Ren and Russell. In contrast to the work by Dorfi & Drury the moving mesh equation is derived directly from the equidistribution principle. In several moving mesh partial differential equations (MMPDE's) are derived in this manner, with the aims of the resulting algorithm being simple, easy to program and relatively insensitive to the choice of user-defined parameters. In all seven of these MMPDE's are constructed using three different approaches, the first two of which are motivated by equidistribution. Using the one-dimensional computational and physical coordinate systems as described in Section 2.1 two quasistatic equidistribution principles (QSEP's), are obtained by differentiating the integral form of the equidistribution principle

(2.3) with respect to ξ once and twice respectively.

$$M(x(\xi, t), t) \frac{\partial}{\partial \xi} x(\xi, t) = \int_0^1 M(\tilde{x}, t) d\tilde{x} \quad (2.20)$$

and

$$\frac{\partial}{\partial \xi} \left\{ M(x(\xi, t), t) \frac{\partial}{\partial \xi} x(\xi, t) \right\} = 0. \quad (2.21)$$

To introduce node movement into the picture, time differentiation is undertaken. Several mesh movement equations have been produced by, for example Anderson Hindman & Spencer and Ren & Russell the former two papers being early attempts with the transformation between physical and computational space, first in one and later in two

dimensions however some of these earlier forms include time differentiation of the integral quantity

$$\theta(t) = \int_0^1 M(\tilde{x}, t) d\tilde{x}.$$

Huang, Ren & Russell state, without supporting

argument, that the quantity $\theta(t)$ or its time derivatives are too complicated to include in actual computation. However, by first differentiating the original equidistribution principle with time and then with, twice we obtain

$$\frac{d}{dt} \left\{ \frac{\partial}{\partial \xi} \left(M \frac{\partial x}{\partial \xi} \right) \right\} = 0$$

which can be written as (MMPDE1)

$$\frac{\partial}{\partial \xi} \left(M \frac{\partial \dot{x}}{\partial \xi} \right) + \frac{\partial}{\partial \xi} \left(\frac{\partial M}{\partial \xi} \dot{x} \right) = - \frac{\partial}{\partial \xi} \left(\frac{\partial M}{\partial t} \frac{\partial x}{\partial \xi} \right) \quad (2.22)$$

so giving a moving mesh equation without reference to $\theta(t)$. In the same paper an alternative set of moving mesh equations, MMPDE's 2-4 are derived by considering (2.21) and requiring that the mesh satisfy the condition at the later time $t + \tau$ (where $0 < \tau \ll 1$) instead of at time t i.e.

$$\frac{\partial}{\partial \xi} \left\{ M(x(\xi, t + \tau), t + \tau) \frac{\partial}{\partial \xi} x(\xi, t + \tau) \right\} = 0.$$

This equation is thought to be a strong enough condition to regularize the mesh movement by Huang et al. Substituting the expansions

$$\begin{aligned} \frac{\partial}{\partial \xi} x(\xi, t + \tau) &= \frac{\partial}{\partial \xi} x(\xi, t) + \tau \frac{\partial}{\partial \xi} \dot{x}(\xi, t) + O(\tau^2) \\ u(x(\xi, t + \tau), t + \tau) &= u(x(\xi, t), t) + \tau \dot{x} \frac{\partial}{\partial x} u(x(\xi, t), t) \\ &\quad + \tau \frac{\partial}{\partial t} u(x(\xi, t), t) + O(\tau^2) \end{aligned}$$

into (2.2) and dropping higher order terms gives MMPDE 2 (2.23) which in fact is MMPDE1 with an additional 'correction' term

$$\frac{\partial}{\partial \xi} \left(M \frac{\partial \dot{x}}{\partial \xi} \right) + \frac{\partial}{\partial \xi} \left(\frac{\partial M}{\partial \xi} \dot{x} \right) = - \frac{\partial}{\partial \xi} \left(\frac{\partial M}{\partial t} \frac{\partial x}{\partial \xi} \right) - \frac{1}{\tau} \frac{\partial}{\partial \xi} \left(M \frac{\partial x}{\partial \xi} \right) \quad (2.23)$$

The extra term is a measure of how well the current grid is equidistributed and hence MMPDE 2 moves the grid towards an equidistributed state even when M is independent of t . For this reason, terms involving $\frac{\partial M}{\partial t}$ are less important for MMPDE 2 than MMPDE 1 and disregarding these terms leads to MMPDE's 3 and respectively, i.e.

$$\frac{\partial^2}{\partial \xi^2} (M \dot{x}) = \frac{1}{\tau} \frac{\partial}{\partial \xi} \left(M \frac{\partial x}{\partial \xi} \right) \quad (2.24)$$

and

$$\frac{\partial}{\partial \xi} \left(M \frac{\partial \dot{x}}{\partial \xi} \right) = \frac{1}{\tau} \frac{\partial}{\partial \xi} \left(M \frac{\partial x}{\partial \xi} \right). \quad (2.25)$$

The remaining MMPDE's (5-7) are devised by considering attraction and repulsion pseudo-forces between nodes. Here the mesh movement is specifically motivated by taking the monitor to be some error measure, so nodes are attracted together when the error is larger than average and repelled when the measure is below average. The error is then expressed as an integral over each cell, W_i , usually taking the form

$$W_i = \int_{x_i}^{x_{i+1}} M(\tilde{x}, t) d\tilde{x}.$$

MMPDE's (5-7) stem from this relation and all involve the correction term mentioned above, which seems to be a key term as it can determine the time-scale for the mesh movement and hence can be adapted to suit the problem in hand. Moreover since the correction term can be derived from the equidistribution idea, its inclusion in the latter mesh equations suggests that the error is evenly distributed over the mesh and the equidistribution and attraction/repulsion ideas are therefore thought to be closely related. Huang, Ren & Russell also provide theoretical analysis suggesting that the MMPDE's cannot produce instances where nodes cross paths when the MMPDE is solved exactly, indicating stability of the resulting meshes. The stability analysis follows early work by Flaherty et al. In particular it is noted that for MMPDE 1 the mesh would be stable if the measure

$$L(t) = \max_{0 \leq \xi \leq 1} \frac{M(x(\xi, 0), 0)}{M(x(\xi, t), 0)}$$

were to remain bounded. However for most choices of M , $L(t)$ is likely to increase, Li et al, went on to discuss the stability of such moving mesh systems in greater detail.

A so-called Moving Mesh Differential Algebraic Equation (MMDAE) is developed by Mulholland, Qiu & Sloan. Instead of using the an MMPDE, the mesh movement is prescribed by a QSEP (2.20 & 2.21) written in terms of an algebraic equation involving the stationary grid points and the monitor function M . In fact the algebraic relation is the equidistribution relation written previously in Section 2.1 equation (2.6). This is coupled with the moving grid Lagrangian form of the underlying PDE and

integrated forward in time using a first-order backward Euler method (used since these systems tend to be stiff). In this technique is used in conjunction with a pseudo-spectral processing of the solution of hyperbolic problems, Qiu & Sloan continue the work, comparing the method and in particular the stability with the established MMPDE 5 of Huang et al. Of particular interest is the stability of the discrete solution of the steadystate solution to Burgers, equation by examining possible steady solutions arising from the two adaptive discretisations of the unsteady problem.

MOVING MESH METHOD IN ONE-DIMENSION

In this chapter we present a moving mesh method to take care of these difficulties. The method stems from observations on equidistribution and geometric conservation laws. In the following sections we shall follow the development of the method from its initial state equidistributing integrals of mass through to the introduction of more complex monitor functions. All stages of the development of the algorithm are supplemented and illustrated with numerical results. The method is presented first in one-dimension and with explicit reference to the PME.

Mass Conservation and Equidistribution

For instance, Qiu & Sloan developed a specialist monitor (2.26) for the solution of a reaction diffusion type problem where more traditional monitors failed. Of particular interest when considering the PME is the work of Budd et al whose choice of monitor was heavily influenced by the theoretical invariance properties of the underlying PDE. In particular the

mass monitor, $M(u) = u$, was singled out as a sensible choice for the PME since it too was invariant under scaling and would conserve mass.

Seizing upon this, it is noted that by choosing this monitor we can derive a moving mesh method without the need for the use of a computational or reference grid. The resulting mesh x and solution u can then be coupled in such a way that only the mesh needs to be integrated forward in time with the solution being recovered from the current grid positions together with an integral quantity relating to the chosen monitor.

1. A Moving Mesh Equation

We begin with a simple equidistribution principle, working on a grid comprising of $N + 1$ nodes $x_0, \dots, x_N$ and, using notation in line with the existing literature, we have that

$$\int_{x_i(t)}^{x_{i+1}(t)} u dx = \theta(t) = \frac{1}{N} \int_{x_0(t)}^{x_N(t)} u dx \quad i = 0, \dots, N-1. \quad (5.1)$$

As previously noted, many moving mesh methods are derived from introducing node speeds into existing grid

adaption statements. When considering the PME with its mass conservation property direct time differentiation of the above equidistribution rule (5.1) leaves us with the simple expression.

$$\int_{x_i(t)}^{x_{i+1}(t)} u_t dx + u(x_{i+1}, t) \dot{x}_{i+1} - u(x_i, t) \dot{x}_i = \theta_t. \quad (5.2)$$

Substituting in the PME (2.30) and simplifying the integral term on the left-hand side leaves us with the moving mesh equation.

$$[u^m u_x]_{x_i(t)}^{x_{i+1}(t)} + u(x_{i+1}, t) \dot{x}_{i+1} - u(x_i, t) \dot{x}_i = 0. \quad (5.3)$$

It is crucial to note, with this specific choice of monitor $M(u) = u$, that θ becomes independent of time since the PME is mass conserving. Hence, the zero right hand side in (5.3). Moreover, due to the symmetry of our porous medium solution, we have that $\dot{x}_0 = \dot{x}_N = 0$, so the above equation, when rearranged, leads to the sequence of ordinary differential systems (ODEs) for the grid co-ordinates.

$$\begin{aligned} \dot{x}_0 &= 0, \\ \dot{x}_i &= \frac{1}{u_i} (u_{i-1} \dot{x}_{i-1} - [u^m u_x]_{x_{i-1}(t)}^{x_i(t)}) \quad i = 1, \dots, N. \end{aligned} \quad (5.4)$$

To discretise the system we use an upwinding approximation for the space derivative terms, i.e.

$$u^m u_x|_{x_i} \approx \left[\frac{u_i + u_{i-1}}{2} \right]^m \frac{(u_i - u_{i-1})}{(x_i - x_{i-1})} = u_{i-\frac{1}{2}}^m \left(\frac{u_i - u_{i-1}}{x_i - x_{i-1}} \right). \quad (5.5)$$

We choose this style of discretisation, since upon expansion the PME can be written in a hyperbolic form for which an upwind approximation is deemed suitable, namely

$$u_t = m u^{m-1} u_x^2 + u^m u_x.$$

Obviously our solution is not complete, since we have not yet stated how to evaluate the values of u at the current grid positions. The next section shows how the solution can be obtained from the current state of the mesh and a discrete integral of u relating to the original equidistribution idea.

2. The Porous Medium Solution

The moving mesh equation presented above (5.4 with 5.5) has already coupled together the prescription for the motion of the grid and the dynamics of the PME. Due to (5.2) the resulting system of ordinary differential equations should move the grid in such a way that, as the material is diffused, computational cells hold an equal quantity

of mass. Restating the equidistribution principle above, our grid should then, at all times, satisfy

$$\int_{x_i(t)}^{x_{i+1}(t)} u dx = \frac{1}{N} \int_{x_0(t)}^{x_N(t)} u dx = \theta \quad i = 0, \dots, N-1.$$

Using a trapezium rule approximation for the equidistributed mass $\theta(t)$ we have that a piecewise linear approximation will satisfy

$$\frac{1}{2}(u_{i+1} + u_i)(x_{i+1} - x_i) = \theta, \quad (5.6)$$

Since $u = 0$ at the foot of the moving boundary we can rearrange equation (5.6) to give us a sequence of algebraic equations yielding the approximate solution u in terms of the current grid co-ordinates and the constant mass θ .

$$\begin{aligned} u_N &= 0, \\ u_i &= \frac{2\theta}{(x_{i+1} - x_i)} - u_{i+1} \quad i = N-1, \dots, 0. \end{aligned} \quad (5.7)$$

Alternatively these algebraic relations can be written explicitly for u in summation form as

$$\begin{aligned} u_N &= 0, \\ u_i &= 2 \sum_{k=i+1}^N \theta \frac{(-1)^{k-i-1}}{x_k - x_{k-1}} \quad i = N-1, \dots, 0. \end{aligned}$$

Some compensation has to be made between the exact conserved equidistributed mass as used when deriving the system (5.4) and the discrete conserved mass θ used above. However if we start with a grid that has equidistributed discrete mass such that (5.6) holds overall cells, then appropriate discretisations of the ODE system will move the grid in such a way that this discrete approximation to the mass will be equidistributed and conserved. This is easily achieved by using a linearised form of the monitor function when generating the initial grid to complement the trapezium rule approximation used in (5.6). We now proceed to deal with the moving boundary involved in the PME, in particular correctly approximating the speed of the moving front.

MOVING MESH METHOD FOR FURTHER APPLICATIONS

Whilst the principles of the mesh movement are viable when tackling other problems, the existence of a known solution value, e.g. $u = 0$, at the moving boundary in the PME allowed us to recover the global solution directly from the calculated grid via an algebraic equation. The method also took advantage heavily of the mass conservation properties of the equation. For these reasons the method could be seen to be too problem specific and lacking in robustness. The study illustrates how the method can be extended

to solve problems without these inherent properties. The equation does conserve mass but has a Neumann boundary condition imposed at the fixed boundaries, hence we do not know a value of u at any of these points from which to construct our resulting solution.

The SemiConductor Problem Revisited -

It was concluded in Chapter 4 that for a numerical method to adequately solve the semiconductor problem the method itself must conserve mass exactly. With this in mind the dynamic grid method introduced in the previous chapter seems to suit his application perfectly. However, the solution will now have a time dependent solution value at both boundaries. When considering the PME the zero value of u at the moving boundary permitted an obvious computational saving since the solution could be derived directly from the grid

and the quantity $\theta(t)$ via an algebraic relation (see Sections 5.1.2, 5.3.1 and 5.4), without a separate time integration for the solution u . Ideally we would like to retain this characteristic of the existing algorithm.

Solution Technique - We begin by recalling the model semiconductor problem. Defined on the fixed region $x \in [0, 1]$, the diffusion of a dopant through silicon is modelled by the equation

$$u_t = ((u + \epsilon)u_x)_x, \quad (6.1)$$

ϵ being a constant of the order $10^{\frac{1}{2}}$. Neumann conditions are imposed at each boundary, $x = 0$ and $x = 1$, and the dopant has initial Gaussian distribution

$$u(x, 0) = e^{-\sigma x^2}.$$

Previously we have chosen σ to have a value of 50 and we continue to do so.

The combination of a fixed domain and Neumann boundary conditions force u to fluctuate away from a value of zero at $x = 1$. In order for our method to allow us to trace back from this point and form the solution from the grid, we need to have a handle on this value u at this point. Due to the low concentration and gradient at this region of interest, we propose to make use of the resulting low temporal changes in u at this boundary and implement a local explicit finite-difference solution. We begin by deriving an expression for the time derivative of u at this point

We shall again be using a grid consisting of $N+1$ nodes numbered $x_0, \dots, x_N$ where $x_0 = 0$ and $x_N = 1$. With reference to Section 3.2.1. Chapter 3, we consider the integral form of equation 6.1 in the

localised region of the control volume of node x_N . As before this region is formally defined as the length between x_N and the cell midpoint $x_{N-\frac{1}{2}}$.

Given the Neumann conditions imposed at the right hand boundary we have that

$$\begin{aligned} \int_{x_{N-\frac{1}{2}}}^{x_N} u_t dx &= \int_{x_{N-\frac{1}{2}}}^{x_N} ((u + \epsilon)u_x)_x dx \\ &= -(u + \epsilon)u_x|_{x_{N-\frac{1}{2}}} \end{aligned}$$

Discretising in an upwind manner as before we derive an approximate expression for u_t via

$$\begin{aligned} (x_N - x_{N-\frac{1}{2}})u_t &= -(u_{N-\frac{1}{2}}^n + \epsilon) \left[\frac{u_N - u_{N-1}}{x_N - x_{N-1}} \right], \\ u_t &= -2(u_{N-\frac{1}{2}}^n + \epsilon) \left[\frac{u_N - u_{N-1}}{(x_N - x_{N-1})^2} \right]. \end{aligned} \quad (6.2)$$

Despite the low gradient of the solution at $x = 1$, a reasonable mesh spacing will always be required near that point during the solution to ensure accuracy of the value u_N and hence the global solution. Taking this into consideration, we propose a further modification to the combination monitor used in the previous chapter. We now need to ensure that the mesh is adequately represented over the regions of low concentration that initially lies away from the Gaussian maximum value. If we consider the combination monitor used previously it is obvious that the value of the monitor will diminish quickly in regions of low concentration.

Numerical Results - We shall analyse the performance of the moving mesh method in this application by computing an approximate solution on a stationary regular mesh consisting of 100001 nodes. The solution is generated using a semi-implicit scheme with an adaptive time-stepping approach, as outlined for the solution of this problem in Chapter 3. We have computed these stationary mesh solutions for three different values $\epsilon = 0.5, 0.01, 0.005$ for comparison with the moving mesh solutions. Before we look at the resulting solutions and the effect of adjusting the leakage of material in the concentration profile, we turn our attention to the error in the semiconductor solution with reference to our fine scale computation.

CONCLUSION

Grid adaption and the use of moving meshes has evolved dramatically over recent years, becoming an essential tool in the successful numerical solution to a wide variety of applications. The ability of a mesh to automatically adjust its distribution in order to resolve steep or sharp solution variations can aid the numerical analyst in gaining effective control over computational resources.

This study has illustrated two contrasting moving mesh methods for the solution of parabolic PDEs. To be more precise, deficiencies found when computing numerical solutions using an initial static method have motivated the development of a dynamic approach which was able to resolve difficulties found in the application of the former algorithm. This final chapter serves as a summary of the work presented and suggests possible future avenues of study.

REFERENCES

1. C.J. Budd and G. Collins (2006). An invariant moving mesh scheme for the non-linear diffusion equation. Technical Report 19/08/96, School of Mathematics, University of Bath.
2. D.A. Anderson (2003). Adaptive mesh schemes based on grid speeds, A.I.A.A. 1: pp. 1311.
3. G.Beckett, J.A. Mackenzie, J. Ramage, and D.M. Sloan (2000). On the numerical solution of one-dimensional partial differential equations using adaptive methods based on equidistribution. Technical Report 2000/02. Strathclyde Mathematics Research Report.
4. G. Beckett, J.A. Mackenzie, and M.L. Robertson (2009). A moving mesh finite element method for the solution of two-dimensional stefan problems. Technical Report 99/26. Dept of Mathematics, University of Strathclyde.
5. J.G. Blom. J.M. Sanz Serna. and J.G. Verwer (2008). On simple moving grid methods for one-dimensional evolutionary partial differential equations, Journal of Computational Physics, 74: pp. 191-213.
6. J.A. Mackenzie and M.L. Robertson (2000). A moving mesh method for the solution of the one-dimensional phase-field equations. Technical report, Dept of Mathematics, University of Strathclyde.
7. M.J. Baines (2009). Least squares and approximate equidistribution in multi dimensions. Numerical Methods Partial Differential Equations, 15: pp. 605-619.
8. R.M. Furzeland. J.G. Verwer. and P.A. Zegeling (2000). A numerical study of three moving grid methods for one-dimensional partial differential equations which are based on the method of lines. Journal of Computational Physics, 89: pp. 349-388.

9. S.Li. L. Petzold and Y. Ren (2006). Stability of moving mesh systems of partial differential equations. SIAM Journal of Scientific Computing, 20: pp. 719-738.
10. W. Cao. W. Huang and R.D. Russell (2010). An r-adaptive finite element method based upon moving mesh partial differential equations. Journal of Computational Physics, 149: pp. 221-244.
11. W. Haung (2012). Practical aspects of formulation and solution of moving mesh partial differential equations. Technical Report 99-11-02, University of Kansas.
12. W. Huang. Y. Ren and R.D. Russell (2014). Moving mesh methods based on moving mesh partial differential equations. Journal of Computational Physics, 113: pp. 279-290.
13. W. Huang and R.D. Russell (2011). Adaptive mesh movement - the MMPDE approach and its applications. Journal of Computational and Applied Mathematics, 128: pp. 383-398.

Corresponding Author

Sikander*

Assistant Professor of Mathematics, A.I.J.H.M. College
Rohtak, Haryana, India

sikanderbeniwal2212@gmail.com