Chapter 2

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 (MMPDE’S) presented by Huang, Ren and Russell [44]. 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 [47], [48].

In the first section of this chapter we shall explore some of the existing ideas in static grid generation so as to give a good understanding of the aims and methods behind many moving mesh methods. We then continue, by following how many of these methods are used to derive moving mesh techniques in one and two dimensions. The last section in the chapter details recent work in which moving mesh methods are chosen in accordance with theoretical properties of the underlying PDE.
2.1 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, first introduced by De Boor [23] for obtaining good discrete approximations to continuous functions. The principles of the method were later applied to generating efficient computational grids for the numerical solution of steady PDEs. For example White [74] used a transformation to arc-length coordinates to generate equidistributing meshes for the numerical solution of two-point boundary value problems. Another approach is given in Denny & Landis [26], 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 define 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,

$$ M = M(x, u, u_x, u_{xx}). $$  \hfill (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 $x_i, i = 0, \ldots, N$, such that $x_0 = a$ and $x_N = b$, the equidistribution principle can be written

$$ \int_{x_i}^{x_{i+1}} Mdx = \frac{1}{N} \int_{a}^{b} Mdx \quad i = 0, \ldots, N-1. $$  \hfill (2.2)

However, in most grid generation applications it is often more convenient to think of the equidistribution idea as one of a co-ordinate 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. Good explanations of this coordinate transform idea can be found in Baines [3] and Huang, Ren & Russell [44]. 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 $\xi_i$ in the computational domain. Written formally, $x$ is then a mapping from $\xi$ to $x$

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

Within this framework the equidistribution idea is written as

$$\int_0^{x(\xi_i)} M \, dx = \xi_i \int_0^1 M \, dx \quad (2.3)$$

or

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

Differentiating (2.3) with respect to $\xi$ once gives the equation used by White [74], differentiating yet again yields the equation presented in Baines [3]

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

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

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

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

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

which may be discretised in a semi-implicit style as follows,
\[ M(x^p_{i+\frac{1}{2}})(x^{n+1}_{i+1} - x^{n+1}_i) - M(x^p_{i-\frac{1}{2}})(x^{n+1}_i - x^{n+1}_{i-1}) = 0. \] (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.

We now consider a few examples of possible monitor functions. The simplest such monitor, \( M = 1 \), produces an uniform equi-spaced grid. This monitor has been used in a moving mesh method with a moving boundary by Budd et al [16], as it permits attractive theoretical properties of the solution within the mesh movement, (details of which shall be discussed later in Section 2.3). Elsewhere, early work by Carey & Dinh [21] showed that minimising the error between a numerical approximation over a computational cell was equivalent to equidistributing the curvature monitor raised to a specific power, depending on which error norm was considered. However, the most common desired feature of using the technique of equidistribution is that the resulting grids have high mesh resolutions where solution gradients are steep and lower resolutions where the solution is less active. This in turn implies that the grid will then provide good approximations of derivative terms when using a suitable numerical scheme or solver. For this reason it is common for the monitor to involve derivative terms of the solution \( u \). In this case, the simplest idea is to use the first derivative of \( u \) with respect to \( x \), i.e.

\[ M = |u_x|. \] (2.7)

The effect of the gradient monitor on a monotonic function is that the solution values themselves become equi-spaced, since,
\[ \int_{x_i}^{x_{i+1}} M dx = u_{i+1} - u_i \]

see Figure 2.1. The most popular choice of monitor is the arc-length of the solution which has been used in many mesh generation and moving mesh methods (e.g., White [74], Dorfi & Drury [27]). The arc-length monitor is written as

\[ M = \sqrt{1 + u_x^2}. \]  \hfill (2.8)

This monitor gives a smoother mesh overall than the gradient monitor, especially when encountering large variations in \( u \), as shown in Figure 2.1.

![Example of Grids using the gradient monitor (2.7) (left) and the arc-length monitor (2.8) (right).](image)

**Figure 2.1:** Examples of Grids using the gradient monitor (2.7) (left) and the arc-length monitor (2.8) (right).

In practice the derivative term in the arc-length monitor is often scaled by some parameter \( \alpha \), for example

\[ M = \sqrt{1 + \alpha u_x^2}. \]  \hfill (2.9)

We shall comment more on the choice of monitor functions later on in Section 2.3.

Although equidistribution is the most common tool used when generating irregular computational meshes in one-dimension, the principle does not however extend strictly into two-dimensions and an alternative is needed.

One of the earliest, and most celebrated of such grid generation approaches in two dimensions is given in the appendix of Winslow’s paper [77]. The main body
of which contains a method for the solution of a quasi-linear Poisson equation on
a non-uniform triangular mesh, and the accompanying appendix outlines how to
form such a mesh for regular domains. The ideas presented in this paper provide
a basis for many of the higher-dimensional grid generation methods that followed.
Once again, the approach is based on a mapping from a computational domain $\Omega_c$
to a physical domain $\Omega_p$. The computational domain is represented as a regular
equilaterial triangular mesh composed of 2 sets of straight lines, associated with the
inverse mappings $\xi(x, y)$ and $\eta(x, y)$ which satisfy the Laplace equations

$$\nabla^2 \xi = 0, \quad (2.10)$$
$$\nabla^2 \eta = 0. \quad (2.11)$$

The solution to (2.10 and 2.11), results in intersecting equi-potentials, i.e. $\xi =\constant$ and $\eta = \constant$, with the mesh completed using the intersections of the
resulting sets of lines. The required mesh is found by inverting the transforms and
putting them in terms of $x(\xi, \eta)$ and $y(\xi, \eta)$ using the Jacobian $J = x_\eta y_\xi - x_\xi y_\eta$, so
that (2.10 and 2.11) become

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = 0, \quad (2.12)$$
$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = 0 \quad (2.13)$$

where

$$\alpha = (x_\eta^2 + y_\eta^2),$$
$$\beta = (x_\xi x_\eta + y_\xi y_\eta),$$
$$\gamma = (x_\xi^2 + y_\xi^2).$$

These equations can be discretised by the finite difference method outlined in the
main body of the Winslow article [77] and solved via an iterative successive over-
relaxation algorithm. Due to the averaging property of the Laplace equation the
constructed mesh is in some sense smooth and is also easily applicable to quadrilateral meshes. Notice that the method is in no way linked to a function or numerical solution represented on the grid. The purpose of this early grid generation algorithm is to produce grids adapted to a particular domain, the shape of which is imposed via boundary conditions used in conjunction with (2.12 and 2.13). Winslow's method as outlined above was adopted by Thompson et al [70] to generate meshes around multiple curvilinear bodies used in modelling flow over various shaped airfoils.

Brackbill and Saltzman [13] took advantage of the idea and extended the method by allowing discretionary control of various mesh properties such as the smoothness and the orthogonality of the grid. Their paper highlights that solving the Laplace equations (2.10 and 2.11) is equivalent to minimising the functional (2.14) below which relates to the smoothness of the mesh, over the computational domain \(\Omega_c\).

\[
I_s = \int_{\Omega_c} [((\nabla \xi)^2 + (\nabla \eta)^2] dV. \tag{2.14}
\]

Similarly, by solving the Euler equations associated with minimising the functional related to the orthogonality of the mesh (see (2.15)), an orthogonal grid is produced.

\[
I_o = \int_{\Omega_c} (\nabla \xi \nabla \eta)^2 dV. \tag{2.15}
\]

In practice, Brackbill and Saltzman suggested the use of linear combinations of such functionals, with the preferences of the user implemented through choices of coefficients. The over-riding theme seems to be that as such properties can be measured they can also be controlled. In their paper the variational approach was used in conjunction with a numerical solution to a steady PDE and results show that, as the chosen functional is minimised, so too is the numerical error. Hence we see the development of the idea of a choice of functional in higher dimensions mirroring the effect of a monitor function in one-dimensional equidistribution.

The methodology of Winslow [77] and Brackbill & Saltzman [13] can be thought of as special cases of a more general framework outlined later by Huang & Russell [46], [47] and [48]. Specifically, [48] presents the following functional (2.16) as a general form of a grid adaptation functional,
\[ I[\xi, \eta] = \frac{1}{2} \int_{\Omega_p} [\nabla \xi^T G_1^{-1} \nabla \xi + \nabla \eta^T G_2^{-1} \nabla \eta] \, dx \, dy \]  

(2.16)

where \( G_1 \) and \( G_2 \) are given symmetric positive definite matrices, referred to as the monitor functions. The desired mesh transformation is derived from the solution to the associated Euler-Lagrange equations,

\[
\nabla \cdot (G_1^{-1} \nabla \xi) = 0, 
\]

(2.17)

\[
\nabla \cdot (G_2^{-1} \nabla \eta) = 0. 
\]

(2.18)

It is easy to see that by choosing \( G_1 = G_2 = I \) this general methodology reduces to Winslow’s original ideas. Moreover Huang & Russell \cite{48} give forms of \( G_1 \) and \( G_2 \) which correspond to Brackbill’s mesh generation method \cite{12}.

Equations (2.17) and (2.18), together with Dirichlet boundary conditions form a harmonic map from the physical to computational domains and the reliability of the method stems from the guaranteed existence and uniqueness of the transform, provided that the boundary of \( \Omega_p \) is convex and that \( G_1 = G_2 \). Details can be found in Dykinsky \cite{30}. Again, in \cite{46}, details of the specific monitor used in \cite{30} are given, involving distances from a given surface. As an illustrative example, below is the ‘arc-length-like’ monitor presented in \cite{48}

\[
G_1 = G_2 = \frac{1}{\sqrt{1 + ||\nabla u||^2}} (I + \nabla u \nabla u^T). 
\]

(2.19)

Further work by Cao et al \cite{20} proved by the use of Green’s functions that the mesh can be aligned in certain directions and mesh concentrations can also be influenced in certain directions by controlling the eigenvectors and eigenvalues of the monitor matrices (specifically when \( G_1 = G_2 \)). In particular, findings from this paper suggest that minimising the function \( I \) concentrates nodes in regions where the eigenvectors of \( G_1 \), \( \lambda_1 \) and \( \lambda_2 \) change significantly. This seems to have stemmed from earlier work by Brackbill \cite{12} and Knupp \cite{54}. The latter followed his own earlier work, this time combining the Winslow functional and another functional giving a certain amount of directional control over the grid by attempting to align mesh lines.
with a prescribed vector field related to the approximate solution. Knupp [34] also
used the variational approach to grid generation, using weights from sets of vector
fields, with the resulting meshes aligning themselves with the same vector fields in
some least-squares sense, of course some prior knowledge of the appropriate vector
fields being needed.

Another interesting example of the application of Winslow and other such meth-
ods, is outlined in Farmer [32] for use in modelling geological features. Here grids
are needed which honour 'control lines' representing features such as faults. These
control lines are extended to the boundaries of the domain via interpolation, leaving
the domain sectioned into several rectangular domains, which are then discretised
using the outlined grid generation techniques.

This functional framework for finding the desired mesh transformation is a pop-
ular and convenient one, especially when used as the basis of a higher dimensional
moving method, as we shall see later. For completeness, it is worth noticing that in
one dimension, minimising the functional

\[ I[\xi] = \frac{1}{2} \int_0^1 \frac{1}{M} \left( \frac{\partial \xi}{\partial x} \right)^2 dx \]

yields the equidistribution equation for a given monitor \( M \) [48]. Since this general
framework has been developed to work as part of high dimensional moving methods,
solution procedures for these methods incorporating the mesh movement process will
be outlined later in Section 2.2.

Alternative two-dimensional analogues of equidistribution for grid generation can
be found in Baines [3] and Huang & Sloan [50]. In the former paper an equation
to solve for the appropriate monitor function is given as a natural generalisation of
equation (2.5),

\[ \nabla_\zeta (M(n) \nabla_\zeta n) = 0 \]

where \( n \) is a coordinate along the direction of \( \nabla u \) and \( \zeta = (\xi, \eta) \). This translates
into a 'local equidistribution in the direction of \( \nabla u \)'. Replacing \( n \) by \( x \) or \( y \) gives the
equations below, which are of the familiar Euler-Lagrange form presented earlier,
$$\nabla_{\zeta}(M\nabla_{\zeta}x) = 0,$$
$$\nabla_{\zeta}(M\nabla_{\zeta}y) = 0.$$

These equations are again solved with an interleaving approach with Dirichlet conditions. The resulting grid is unable to equidistribute $M$ precisely but clusters grid points in regions of high $M$ as desired. Further, Baines [4] shows that a least squares minimization of a residual of a vector field is equivalent to a least squares measure of equidistribution on triangular meshes, in some sense extending the work in one dimension by Carey & Dinh [21].

Elsewhere, the work of Huang and Sloan [50] follows ideas set out by Dwyer [31] and Catherall [22] and a local equidistribution is obtained by imposing the strict one-dimensional form over two sets of co-ordinate lines.

It is worth taking time to grasp an understanding of these grid generation techniques as a precursor to studying moving-mesh methods. As we shall see in the following section, many moving grid algorithms are based upon an underlying principle for constructing meshes with effective grid resolutions.

### 2.2 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. We shall make constant reference to the techniques in Section 2.1, so it makes sense to follow the same order of events, starting with the use of the equidistribution principle in deriving moving mesh methods in one dimension.

An early incorporation of the equidistribution idea into a moving mesh method is outlined by Petzold [34]. 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 pre-determined error measure, during the forward integration in time. In other words, at certain times throughout the numerical solution of the equation, the grid is re-equidistributed, 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 [11] used a predictive step, re-equidistribute 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.

In contrast to the regridding idea, an early dynamic moving mesh technique was devised by Dorfi & Drury [27]. Here a separate equation for mesh speeds is developed via a function $R$ to control mesh resolution which acts in the same way as a monitor function (despite no formal mention of equidistribution ideas). A simple relation between the speeds of the points $\dot{z}$ and $R$ is solved in conjunction with the underlying PDE. Other early additional moving mesh equations include the work by Adjerid & Flaherty [1], who used a moving mesh equation within a finite-element framework to equidistribute the local discretisation error within the scheme. Petzold [65] followed the regridding approach with a more dynamic moving mesh method, the idea here being that using transformed pseudo-Lagrangian moving mesh co-ordinates, mesh speeds can be chosen so as to minimise the movement of the mesh in the transformed variables, so the solution in these co-ordinates is changing as slowly as possible for an easier numerical solution.

White [75] followed earlier grid generation work in one-dimension by using a moving mesh method based upon the transformation to arc-length type co-ordinates. Applications of early moving mesh methods include the work by Larroueturou [56], working on a flame propagation problem, a single mesh speed being derived for the entire grid, this velocity chosen to preserve thermal energy in the solution, the entire
The grid is then moved as a rigid body. For the reader's interest, a review of some of the earlier moving mesh methods in one-dimension can be found in Hawken et al [40].

We now turn our attention to the work of Huang, Ren and Russell ([44], [63] and [68]). In contrast to the work by Dorf & Drury the moving mesh equation is derived directly from the equidistribution principle. In [44] 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 quasi-static equidistribution principles (QSEP's), are obtained by differentiating the integral form of the equidistribution principle (2.3) with respect to \( \xi \) once and twice respectively,

\[
M(x(\xi, t), t) \frac{\partial}{\partial \xi} x(\xi, t) = \int_0^1 M(\tilde{x}, t) d\tilde{x}
\]

and

\[
\frac{\partial}{\partial \xi} \left\{ M(x(\xi, t), t) \frac{\partial}{\partial \xi} x(\xi, t) \right\} = 0.
\]

To introduce node movement into the picture, time differentiation is undertaken. Several mesh movement equations have been produced by, for example Anderson [2], Hindman & Spencer [41] and Ren & Russell [68], the former two papers being early attempts with the transformation between physical and computational space, first in one, [41], and later in two dimensions [2]. 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 \( \xi \) 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)

\[
\frac{\partial}{\partial \xi} \left( M \frac{\partial x}{\partial \xi} \right) + \frac{\partial}{\partial \xi} \left( \frac{\partial M}{\partial \xi} \frac{\partial x}{\partial \xi} \right) = -\frac{\partial}{\partial \xi} \left( \frac{\partial M}{\partial t} \frac{\partial x}{\partial \xi} \right)
\]

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

\[
\frac{\partial}{\partial \xi} x(\xi, t+\tau) = \frac{\partial}{\partial \xi} x(\xi, t) + \tau \frac{\partial}{\partial \xi} \dot{x}(\xi, t) + O(\tau^2)
\]
\[
u(x(\xi, t+\tau), t+\tau) = u(x(\xi, t), t) + \tau \frac{\partial}{\partial x} u(x(\xi, t), t)
\]
\[
+ \tau \frac{\partial}{\partial t} u(x(\xi, t), t) + O(\tau^2)
\]

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

\[
\frac{\partial}{\partial \xi} \left( M \frac{\partial x}{\partial \xi} \right) + \frac{\partial}{\partial \xi} \left( \frac{\partial M}{\partial x} \frac{\partial x}{\partial \xi} \right) = -\frac{\partial}{\partial \xi} \left( \frac{\partial M}{\partial t} \frac{\partial x}{\partial \xi} \right) - \frac{1}{\tau} \frac{\partial}{\partial \xi} \left( \frac{\partial M}{\partial \xi} \frac{\partial x}{\partial \xi} \right)
\]

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 4 respectively, i.e,
\[ \frac{\partial^2}{\partial \xi^2} (M \dot{x}) = \frac{1}{\tau} \frac{\partial}{\partial \xi} \left( M \frac{\partial x}{\partial \xi} \right) \]  
(2.24)

and

\[ \frac{\partial}{\partial \xi} \left( M \frac{\partial \dot{x}}{\partial \xi} \right) = \frac{1}{\tau} \frac{\partial}{\partial \xi} \left( M \frac{\partial x}{\partial \xi} \right). \]  
(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

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

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 [33]. In particular it is noted that for MMPDE 1, the mesh would be stable if the measure

\[ I(t) = \max_{0 \leq \xi \leq 1} \frac{M(x(\xi, 0), 0)}{M(x(\xi, t), 0)} \]

were to remain bounded. However for most choices of \( M \), \( I(t) \) is likely to increase. Li et al [58] went on to discuss the stability of such moving mesh systems in greater detail.
The resulting equations (MMPDE’s 1-7) have spawned a variety of work in various applications, sometimes with a common modification, that being the spatial smoothing of the monitor function $M$. Dorfi & Drury [27] and Furueland et al [34] came to the conclusions in their early moving mesh work that when using finite-difference schemes to approximate derivative terms, in order to obtain ‘reasonable’ accuracy the mesh should be, in some sense, smoothed. Verwer et al [72] proved that smoothing the mesh is equivalent to smoothing the monitor function over the grid. Motivated by this work, Huang, Ren & Russell [43] use MMPDE’s (3-7) with a smoothed monitor function $\hat{M}$ defined at each node by

$$\hat{M}_i = \sqrt[\frac{1}{1+p}]\sum_{k=x-p}^{x+p} \left( \frac{M_k}{1+\gamma} \right)^{k-1}$$

where $\gamma$ is a smoothing parameter and $p$ is a non-negative integer referred to as the smoothing index which determines the range of the smoothing. These ideas provide a valuable tool in higher dimensions, since using a locally smoothed monitor function is considerably easier than smoothing the entire mesh separately. Moreover it is noted in [44] and [43] that MMPDE’s 3 and 4 permit a possible extension to multi-dimensions.

Mackenzie [59] and Stockie et al [69] have both applied the smoothed moving mesh equations to PDEs in one-dimension and later to systems of hyperbolic conservation laws, where monitors were not only smoothed but combined to provide a moving grid on which to simulate the development of several time dependent variables. Mackenzie & Robertson [60] also used a mesh equation based upon equidistribution applied to a problem involving a phase change. Here a monitor based upon the asymptotic behaviour of the problem was used, clustering nodes around the moving interface, whilst the inclusion of a constant term also allowed sufficient nodes to be placed away from the region. Further applications of the MMPDE’s (1-7) include work by Qiu & Sloan [67], who applied MMPDE 6 with the outlined technique of smoothing the monitor to Fisher’s Equation. Interestingly enough, a new monitor was derived for specific use with reaction-diffusion problems (2.26) after arc-length and curvature monitors proved to be unsuccessful: this was
\[ M(x, t) = \left[ 1 + \alpha^2(1-u)^2 + \beta^2(a-u)^2 \left( \frac{\partial u}{\partial x^2} \right)^2 \right]^{\frac{1}{2}} \tag{2.26} \]

where \( \alpha, \beta \) and \( a \) are user defined parameters.

Huang and Russell [45], also investigated the addition of artificial diffusion terms to the monitor as a means of smoothing, the resulting method satisfying a mesh crossing condition and allowing for possible extension to higher spatial dimensions.

A so-called Moving Mesh Differential-Algebraic Equation (MMDAE) is developed by Mulholland, Qiu \& Sloan [63]. Instead of using the an MMPDE, the mesh movement is prescribed by a QSEP (2.20 and 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 [63], this technique is used in conjunction with a pseudo-spectral processing of the solution of hyperbolic problems. Qiu \& Sloan [67] continue the work, comparing the method and in particular the stability with the established MMPDE 5 of Huang et al [44]. Of particular interest is the stability of the discrete solution of the steady-state solution to Burgers’ equation by examining possible steady solutions arising from the two adaptive discretisations of the unsteady problem.

We now move on to moving mesh methods in higher dimensions. In the previous section we outlined a class of stationary grid adaptation methods based upon minimising a mesh generation functional. As with the moving-mesh techniques in one dimension, we introduce mesh speeds into such a grid adaption method so as to preserve the properties of the grid as it moves in time. A popular way to introduce mesh speeds into the mesh functional approach is by use of the so-called gradient flow equations. Following the approach of Huang \& Russell [47], a functional \( I[\xi, \eta] \) is minimised over the computational domain \( \Omega_c \). One way to minimise \( I \) is to follow the steepest descent direction given by the first derivative of \( I \). The following ‘gradient flow’ equations define a flow which converge to the equilibrium state at \( t \to \infty \),

21
\[
\frac{\partial \xi}{\partial t} = \frac{\partial I}{\partial \xi}, \quad \frac{\partial \eta}{\partial t} = \frac{\partial I}{\partial \eta}.
\]

In practice a modified version of these equations is used in [47], with the inclusion
of the familiar correction term \( \tau \) and the introduction of \( P \), an operator on the
underlying function space.

\[
\frac{\partial \xi}{\partial t} = -\frac{P}{\tau} \frac{\partial I}{\partial \xi}, \quad \frac{\partial \eta}{\partial t} = -\frac{P}{\tau} \frac{\partial I}{\partial \eta}.
\]

The extra term \( P \) is used to choose more suitable directions than that of steepest
descent with the \( \tau \) terms allowing the user to choose a suitable time scale for the
problem. It has already been noted in Section 2.1 that the functional approach
in one-dimension can be shown to be equivalent to the equidistribution principle.
Moreover the approach here can be shown to be similar to using MMPDE 5 [44],
being based on the attracting and repellent forces of the monitor function. Indeed
Beckett et al [6] used a similar version of the monitor outlined previously (2.26) in
conjunction with a one-dimensional analogue of (2.29) for the solution of Burgers’
equation. More recently MMPDE 5 has been used in two dimensions as part of an
adaptive finite element method by Cao et al [19] for the solution of a combustion
problem consisting of coupled non-linear reaction-diffusion equations.

Huang & Russell give multi-dimensional generalisations using this methodology
for MMPDE’s 4 and 6. Using this approach and the general grid generation functional
(2.16), a suitable \( P \) is given in terms of the determinants of the two monitor
matrices, i.e. \( \hat{g}_1 = det(G_1) \) and \( \hat{g}_2 = det(G_2) \), giving the resulting MMPDE

\[
\frac{\partial \xi}{\partial t} = \frac{1}{\tau \sqrt{\hat{g}_1}} \frac{\partial I}{\partial \xi}, \quad \frac{\partial \eta}{\partial t} = \frac{1}{\tau \sqrt{\hat{g}_2}} \frac{\partial I}{\partial \eta}
\]  

(2.27)
or

\[
\frac{\partial \xi}{\partial t} = -\frac{1}{\tau \sqrt{\hat{g}_1}} \nabla \cdot (\hat{G}_1^{-1} \nabla \xi), \quad \frac{\partial \eta}{\partial t} = -\frac{1}{\tau \sqrt{\hat{g}_2}} \nabla \cdot (\hat{G}_1^{-1} \nabla \eta).
\]  

(2.28)
As with solving for a stationary mesh, the actual computations are carried out after interchanging dependent and independent variables, giving

\[
\frac{\partial \mathbf{\xi}}{\partial t} = -\frac{\mathbf{\xi}_\xi}{\tau \sqrt{g_1 J}} \left\{ \mathbf{\xi}_\eta \frac{\partial}{\partial \mathbf{\xi}} \left[ \frac{1}{J g_1} (\mathbf{\xi}_\eta^T G_1 \mathbf{\xi}_\eta) \right] - \mathbf{\xi}_\eta \frac{\partial}{\partial \mathbf{\xi}} \left[ \frac{1}{J g_2} (\mathbf{\xi}_\eta^T G_2 \mathbf{\xi}_\eta) \right] \right\}
\]

where \( J \) is the Jacobian of the co-ordinate transform.

Given this general framework, equivalent MMPDE’s can be constructed using the various specific functionals described in Section 2.1. Dirichlet boundary conditions are preferred for the solution of (2.29) as this yields a unique solution, but for many problems this is not applicable since the boundary may not be stationary. Indeed, in some cases it is useful to moves nodes around the fixed boundary, for which many techniques are under investigation, the most popular being preserving a one-dimensional arc-length equidistribution of nodes on the boundary (see Huang & Russell [39], [49], Beckett et al [7]).

Huang and Russell [47] outline a familiar interleaving approach for the solution of the higher dimensional MMPDE combined with the underlying physical PDE as follows.

- Calculate the monitor functions \( G_1 \) and \( G_2 \) on the current mesh.

- Update the mesh at time \( t + \Delta t \) by integrating the MMMPDE(2.29), keeping \( G_1 \) and \( G_2 \) constant.

- Integrate the physical PDE to get the solution at time \( t + \Delta t \) using the mesh

\[
\mathbf{x}(t) = \mathbf{x}^n + \frac{(t-t_{n+1})}{\Delta t_n} (\mathbf{x}^{n+1} - \mathbf{x}^n)
\]

and mesh speed

\[
\mathbf{\dot{x}}(t) = \frac{(\mathbf{x}^{n+1} - \mathbf{x}^n)}{\Delta t}.
\]

- Choose a value of \( \Delta t_{n+1} \) for the next time step from the physical PDE.
As with their work in one-dimension, Huang, Ren & Russell suggest that the time correction term \( \tau \) is preset by the user or determined by the development of the solution. However the choice of this value in one-dimension is relatively insensitive and it is thought to be so in higher dimensions also. Central finite difference discretisations are used by Huang & Russell along with a simple rectangular uniform reference mesh for the computational space. Again, extending the work carried out in one-dimension, the monitor is smoothed locally.

On reflection, the functional framework for multi-dimensional moving mesh methods gathers together all of the work described, both in grid adaption and one-dimensional moving grid techniques, since the strict equidistribution ideas in one-dimension can be written in terms of a functional and the moving mesh methods in higher dimensions are derived from a functional approach to grid adaption.

As an interesting aside, work by Demirdžić & Perić, [25] and [24] considers moving mesh methods from a more practical aspect. The authors suggest that many of the moving mesh algorithms before them induce error by not satisfying exactly any relevant conservation laws. Work is continued in [24] and mesh movement equations are derived for the solution of the Navier Stokes equations from a general scalar quantity conservation law. The fact that relevant physical quantities are conserved almost ‘by construction’ in the method is considered to be of utmost importance and is the driving force behind the moving grid.

Whichever approach is undertaken, a good understanding of the numerical techniques alone may not be good enough for the solution of some problems. We shall continue in the next section by introducing recent work which combines moving mesh methods and self-similar solution techniques, which suggests reasonable choices of monitor functions for certain problems. In particular we shall consider application to the solution of the PME, which we now describe in detail.

2.3 Self Similar Solutions and Mesh Movement

Recently there has been great interest in the connection between moving mesh methods and the numerical approximation of solutions to self-similar problems. The main
focus of attention has been on the PME in one-dimension (2.30) and the question of using appropriate moving mesh methods to suit its properties. Most of this work has been carried out by Budd et al. ([15], [16], [18]). In this section we shall outline the main ideas behind the self-similar solutions of the PME and review the resulting moving mesh methods specifically derived for its solution. We begin by formally introducing the PME and highlighting some of its properties.

In one dimension, the PME is a non-linear diffusion equation of the form

$$u_t = (u^m u_x)_x \quad (2.30)$$

where $m > 0$.

Equation (2.30) arises from the study of the diffusion of gas through a porous medium under the action of Darcy’s law relating velocity to pressure gradient. Another application appears in the modelling of the swarming of various insect species from which under certain conditions an analytic solution is available, as presented in Murray [64]. There is now a fairly complete existence theory for the equation, given initial conditions $u(0, x)$, these solutions take the form of travelling waves forming on a region of growing compact support $[s_-(t), s_+(t)]$, with $u = 0$ for $x \geq s_+(t)$ and $x \leq s_-(t)$. The solution conserves two important quantities, namely mass and centre of mass. Given a solution $u$ exists, we have that

$$I(t) = \int_{-\infty}^{\infty} u dx > 0. \quad (2.31)$$

Then

$$\frac{dI}{dt} = \int_{-\infty}^{\infty} u_t dx = \int_{-\infty}^{\infty} (u^m u_x)_x dx = 0.$$ 

Hence $I$, or mass is conserved. Similarly, if $\bar{x}$ is the scaled centre of mass

$$\bar{x} = \int_{-\infty}^{\infty} x u dx,$$

then

$$\frac{d\bar{x}}{dt} = \int_{-\infty}^{\infty} x u_t dx = \int_{-\infty}^{\infty} x (u^m u_x)_x dx = -\int_{-\infty}^{\infty} uu_x dx = -\frac{1}{2} \int_{-\infty}^{\infty} (u^2)_x dx = 0.$$
Since both mass and centre of mass are conserved. The speed of the moving boundaries can be derived from the conservation of mass property. For instance, in the case of \( m = 1 \), the interfaces propagate with finite speed given by

\[
\frac{ds_{\pm}}{dt} = -u(s_{\pm}, t). 
\]

Hence given symmetric initial conditions, the solution will remain symmetric throughout time.

Figure 2.2 illustrates the behaviour of this class of solutions, the dashed line representing the progress of the Murray Solution for \( m = 2 \). The value of \( m \) influences both the speed of the moving boundary and steepness of the solution inside the interface. The larger the value of \( m \) results in a steeper evolving front and a slower rate of displacement.

![Figure 2.2: Evolution of solution to the PME (2.30), for \( m = 2 \).](image)

Moreover the PME has a scaling invariant property which is the basis of the moving mesh ideas of Budd et al. We begin by providing a simple definition.

Given a system \((u, x, t)\) satisfying a PDE, we introduce a mapping to a new system \((\hat{u}, \hat{x}, \hat{t})\) under the transformation

\[
\hat{u} = \lambda^\alpha u, \quad \hat{x} = \lambda^\beta x, \quad \hat{t} = \lambda t \quad \text{(2.32)}
\]
where $\lambda$ is an arbitrary constant. The original system, $(u, x, t)$, is said to be scaling invariant if the PDE under consideration is identical in both the original and transformed coordinates. Moreover, given a solution to the equation, if this too is invariant under the mappings then the solution is said to be self-similar. The benefits of such a transformation being available lies in the fact that sometimes the PDE is easier to solve in the transformed coordinate system. We continue with the derivation of such a transform for the PME.

Using the general form of the transforms (2.32), we can write the left hand side of the PME as

$$u_t = \lambda^{-\alpha} \hat{u}_t$$  \hspace{1cm} (2.33) \\
$$= \lambda^{-\alpha} \hat{u}_t(\lambda)$$ \\
$$= \lambda^{1-\alpha} \hat{u}_t.$$

Now considering the right hand side, substituting the transformations (2.32) gives

$$(u^m u_x)_x = (\lambda^{-m\alpha} \hat{u}^m u_x)_x$$  \hspace{1cm} (2.34) \\
$$= \lambda^{-m\alpha} (\lambda^{-\alpha} \hat{u}^m \hat{u}_x)_{xx}$$ \\
$$= \lambda^{-\alpha(m+1)+\beta} (\hat{u}^m \hat{u}_x)_x$$ \\
$$= \lambda^{-\alpha(m+1)+\beta} \hat{x}_x (\hat{u}^m \hat{u}_x)_x$$ \\
$$= \lambda^{-\alpha(m+1)+2\beta} (\hat{u}^m \hat{u}_x)_x.$$

Equating the two sides (2.34) and (2.35) gives

$$\lambda^{1-\alpha} \hat{u}_t = \lambda^{-\alpha(m+1)+2\beta} (\hat{u}^m \hat{u}_x)_x,$$

and so, comparing the powers of $\lambda$, the system is scaling invariant provided that

$$1 - \alpha = -\alpha(m + 1) + 2\beta$$

or

$$27$$
\[ \alpha m - 2\beta + 1 = 0. \quad (2.35) \]

Furthermore we shall require the behaviour of the solution in the transformed coordinates to mirror the conservation properties of the solution in the original space. Taking into consideration equation (2.31),

\[ I = \int_{-\infty}^{\infty} \lambda^{-\alpha} \hat{u} dx = \lambda^{-(\alpha + \beta)} \int_{-\infty}^{\infty} \hat{u} \hat{x}. \]

So for the mass conservation property to be independent of \( \lambda \), we require that

\[ \alpha + \beta = 0, \]

which in conjunction with (2.35) gives us that the transformation (2.32) must satisfy

\[ \alpha = \frac{-1}{m + 2} \quad \text{and} \quad \beta = \frac{1}{m + 2}. \quad (2.36) \]

Using the above conditions, the equation may be rewritten in terms of the transformed variables and solved to give a class of solutions to the PME. The resulting self-similar solution is invariant and hence independent of time, choosing \( \hat{t} \) arbitrarily to be 1, we can write the transformed solution in terms of the original coordinates as

\[ \hat{u} = ut^{\frac{-1}{m+2}}, \quad \hat{x} = xt^{\frac{-1}{m+2}}. \quad (2.37) \]

The resulting, steady, differential equation can be solved by techniques found in Barenblatt [5] and Dresner[28], which form the basis of the solution given in Murray.

As mentioned above, the Murray solution arises from a model of insect dispersal. The PME comes from the model by assuming that the diffusion of the species population \( Q \) from a central origin is due to population pressure. The exact analytical solution is of the form

\[ u(x,t) = \begin{cases} \frac{1}{\lambda(t)} [1 - \left( \frac{x}{r_0(t)} \right)^2]^{\frac{m}{2}} & |x| \leq r_0\lambda(t) \\ 0 & |x| > r_0\lambda(t) \end{cases} \quad (2.38) \]

where
\[
\lambda(t) = \left( \frac{t}{t_0} \right)^{\frac{1}{2(m+1)}},
\]

\[
r_0 = \frac{Q \Gamma\left(\frac{1}{m} + \frac{3}{2}\right)}{\pi^{\frac{1}{2}} \Gamma\left(\frac{1}{m} + 1\right)}
\]

and

\[
t_0 = \frac{r_{0m}^2}{2(m+2)}
\]

with \(r_0 \lambda(t)\) representing the position of the moving front, and \(\Gamma(x)\) denoting the gamma function.

Upon consideration of the general higher-dimensional form of the PME,

\[
u_t = \nabla . (u^m \nabla u),
\]

Murray also provides a radially symmetric solution in two dimensions. Hence after transforming to radial coordinates, the PME becomes

\[
u_t = \frac{1}{r} (r u^m u_r)_r
\]

and we have the solution

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

where

\[
\lambda(t) = \left( \frac{t}{t_0} \right)^{\frac{1}{2(m+1)}},
\]

\[
r_0^2 = \frac{Q}{\pi} \left( 1 + \frac{1}{m} \right)
\]

and

\[
t_0 = \frac{r_{0m}^2}{4(m+1)}
\]
Similarly there exists a self-similar transformation for this radial case. Using the same forms for the transformation as used previously, we have

\[ \hat{u} = \lambda^{-\beta} u, \quad \hat{r} = \lambda^{-\gamma} r, \quad \hat{t} = \lambda t. \tag{2.41} \]

Upon substitution into the radial form of the PME (2.39), when comparing powers of \( \lambda \), the same relation found in the one-dimensional case is found (2.35). Upon considering the appropriate form for mass conservation, we have that if

\[ I = 2\pi \int_{-\infty}^{\infty} ru dr \]

and

\[
\frac{dI}{dt} = 2\pi \int_{-\infty}^{\infty} ru u dr \\
= 2\pi \int_{-\infty}^{\infty} (ru^m u_r)_r dr \\
= 0.
\]

So, upon substitution of the relevant transforms we have that

\[
I = 2\pi \int_{-\infty}^{\infty} \lambda^{-\beta} \hat{r} \lambda^{-\alpha} \hat{u} dr \\
= 2\pi \lambda^{-(\alpha+\beta)} \int_{-\infty}^{\infty} \hat{r} \hat{u} \lambda^{-\beta} d\hat{r} \\
= 2\pi \lambda^{-(2\alpha+\beta)} \int_{-\infty}^{\infty} \hat{r} \hat{u} d\hat{r}
\]

which, upon comparison of powers of \( \lambda \) gives

\[ \alpha + 2\beta = 0. \tag{2.42} \]

Equations (2.35) and (2.42) give us with the associated self-similar transformation in the radial case, provided that

\[ \alpha = \frac{-1}{m+1}, \quad \beta = \frac{1}{2m+2} \tag{2.43} \]

which, in turn, provides the invariant radial solution in terms of the original coordinates.
\[ \hat{u} = ut^{-\frac{m-1}{m}}, \quad \hat{r} = rt^{-\frac{m-2}{m}}. \]  \hspace{1cm} (2.44)

Although we shall not be directly making use of these transformations in the numerical methods to be introduced in later chapters, the existence of the theory allows us to validate approximate solutions for the PME in both the radial and one-dimensional cases. Our aim is that the approximate solutions will, under the appropriate transform, display the invariant properties of the exact solutions.

Having demonstrated that the true solution to the PME has a self-similar solution under the prescribed transformation, Budd et al conclude that the numerical scheme must inherit the self-similar properties of the analytic solution, specifically a moving mesh should be used for which the monitor is scaling invariant too. In [16] Budd et al implement a moving mesh method involving the simplest of monitors which preserves the said properties, this being \( M = 1 \). Using finite difference approximations to the pseudo-Lagrangian form of the PME with \( m = 1 \) and the speeds of the moving boundary, the resultant mesh has an expanding uniform resolution. The method is applied to the PME in both the original and scaled variables and it is found that the scaling resulting from the discrete self-similar solution is identical to that in the continuous case. Hence the discrete self-similar solution has the dynamics of the underlying solution in the original variables. The scheme used permits the conservation of mass (and the centre of mass) and it is shown that the resulting discrete self-similar solution converges to the true self-similar solution as the number of nodes in the mesh is increased. Later Budd & Piggott [18] suggest that when using the moving mesh PDE to solve scale invariant problems, the monitor should also in some sense be scaling invariant. Another obvious choice is the mass monitor,

\[ M = u, \]

which we shall use later in this thesis. Budd & Piggott’s results show that, when \( m = 1 \) and the evolution of the PME is fairly gentle, the mass monitor preserves all the desired properties and is able to model the shallow front formation behaviour at the moving boundaries. No higher values of \( m \) are considered in [16], for which the front at the moving boundary is considerably steeper and a more suitable monitor
would be needed to resolve the front. Budd et al suggest that, since the monitor is chosen in such a way as to exhibit the properties of the true and self-similar solution, both the discrete solution and the moving mesh method also permit this behaviour.

Moving away from the PME, similar ideas have been applied to the non-linear Schrödinger equation [14] and to problems with blow-up [17]. In the latter, similar results are achieved, even when the smoothing process suggested by Huang, Ren & Russell [43] is applied to the chosen invariant monitor.

It is hoped that this chapter has provided a brief, yet informative, insight into existing techniques for grid adaption and in particular moving mesh methods. Although we shall not be implementing any of these methods directly, some of the ideas explored in this chapter will be of considerable relevance when we come to derive our own techniques as the thesis continues.