C J BUDD AND G J COLLINS

Symmetry based numerical methods for partial differential equations

Abstract We look at numerical methods for differential equations which are invariant under the action of a symmetry group. We show that numerical methods which preserve this symmetry give excellent results when used to compute problems with singularities and with free boundaries

1 Introduction

It has long been recognised that many important partial differential equations arising in mathematical physics and applied mathematics are invariant with respect to a Lie group of transformations and that exact solutions can be determined in many case by exploiting this invariance. Remarkably such solutions are often also global attractors for more general solutions of the equations, given a variety of initial data, and give an accurate indication of the intermediate asymptotic behaviour of the solution after initial effects have died down and before boundary effects become important [2]. Symmetry also plays an important role in studies of ordinary differential equations describing the evolution of systems on manifolds. Loosely speaking, symmetry is what makes interesting equations interesting. It often lies at the heart of the long-time behaviour of the solutions, the nature of any singularities that form, the location and motion of interfaces and the type and stability of pattern formation.

In general, numerical methods are not designed to reflect the symmetry structures of the partial differential equations they are approximating, and this can lead to poor performance in the long-time behaviour of the methods or in the computation of singularities. The original work of Lie, and much subsequent work in applied mathematics, have used symmetry ideas to study solutions of partial differential equations which themselves possess the full symmetry of the differential equation. These generally satisfy a much simpler equation such as an ordinary differential equation, and this latter equation is often solved using a numerical method. However this procedure suffers from disadvantages such as being unable to work with general initial data or boundary conditions, being difficult to set up (the ordinary differential equation may have to be solved over an infinite domain), or being used to compute a solution which does not in fact exist.

In this paper we shall present a class of methods which retain many of the symmetries of the underlying partial differential equation without at the same time being forced into the straight-jacket of having to compute only solutions which are themselves symmetry-invariant. These methods are based on a synthesis of adaptive time stepping with a moving mesh method based on equidistribution. Effectively, they work by letting the numerical method automatically choose the natural variables which govern the evolution of the solution. By considering the two examples of the blow-up equation and the porous medium equation, we shall show their effectiveness in computing singularities, interfaces and long-time behaviour. We conclude the paper with some examples of further problems which are symmetry-invariant and discuss the possible extensions of symmetry-based methods.

2 An introduction to symmetry-invariance in partial differential equations

Symmetry plays such a major role in many differential equations that we cannot hope to present a full treatment here. Good accounts are given in [2], [20], [4], [5] and [12]. In this section we aim to give a short account of some of the basic ideas in the above references. It is worth mentioning that a partial differential equation is often a short-hand description of the underlying physics of a problem, which is often better described in terms of its symmetry groups, and indeed this is often the approach taken by theoretical physicists. Consider then a differential equation of the form

$$u_t = f(u), \quad u(x,t) \in B \tag{2.1}$$

where $f(u)$ is either an algebraic or a differential operator. We can consider both the equation and its solution to be invariant under the action of symmetrey operators.

Invariance of the equation

The equation (2.1) is *symmetry-group invariant* if there exists a group G of transformations of u, x and t which leaves the equation invariant. Typically such a group is a Lie group and the action of the group on u, x and t is expressed in terms of infinitessimal differential operators X_i (see [20]). The simplest way to motivate this is through some examples

Example 1. **Fisher's equation**
This equation, given by

$$u_t = u_{xx} + u(1-u), \quad u \in C^2(-\infty, \infty) \tag{2.2}$$

plays an important role in mathematical biology (studies of genotype transport) and in probability (branching processes). It is invariant under the following groups and associated differential operators X_i.
G_1 : translations in time, $t \to t + \lambda$, $\quad X_1 = \partial_t$.
G_2 : translations in space, $x \to x + \lambda$, $\quad X_2 = \partial_x$.
Z_2 : reflexions in x, $x \to -x$.

Example 2. **The linear heat equation**
The celebrated and universal heat equation

$$u_t = u_{xx}, \quad u \in C^2(-\infty, \infty) \tag{2.3}$$

is invariant under the above groups G_1, G_2, Z_2, and also the *stretching group* G_3 given by

$$t \to \lambda t, \quad x \to \lambda^{1/2} x, \quad X_3 = 2t\partial_t + x\partial_x$$

where $\lambda > 0$ is arbitrary. (In fact, the heat equation is invariant under the action of five further symmetry groups [20].)

Example 3. **The blow-up equation**
This is a nonlinear version of the heat equation, given by

$$u_t = u_{xx} + u^2 \quad u \in C^2(-\infty, \infty). \tag{2.4}$$

It is a model for combusting processes in which materials become hot very quickly. Indeed solutions of (2.4) which have sufficiently large initial data become infinite (forming a singularity) in a finite time. We return to this equation in detail in a later section. It is invariant under the action of the groups G_1, G_2, Z_2 and the stretching group G_4 given by

$$t \to \lambda t, \quad x \to \lambda^{1/2} x, \quad u \to u/\lambda$$

with the associated differential operator $X_4 = 2t\partial_t + x\partial_x - 2u\partial_u$.

Example 4. **Ordinary differential equations on Lie groups**
This is a wide class of problems which are described in detail (together with associated numerical methods) in [16]. In the simplest (linear) case they take the form

$$\dot{\mathbf{u}} = A(t)\mathbf{u} \tag{2.5}$$

where $A \in \mathfrak{g}$ and $\mathfrak{g}$ is the Lie algebra of a Lie group G. This equation is then invariant under the action of G and if u is initially in G, it remains in G during the evolution. For example, $G = \mathrm{O}(n)$ if A is a skew-symmetric matrix. Similarly, $G = \mathrm{SL}(n)$ if A is trace-free.

Solution Invariance

Although an equation itself may be symmetric under the action of a group, there is no guarantee that the solution $u(x, t)$ is itself symmetric. For example, if we take Fisher's equation (2.1) then its solutions need not be invariant under reflexions in x. However, solutions which are also invariant under the action of G play a special role in the study of these equations and are often called *self-similar* solutions.

Many well known solutions of differential equations are self-similar. A solution of Fisher's equation which is invariant under G_1 is the steady-state solution of

$$u_{xx} + u(1-u) = 0.$$

A solution invariant under the action of G_2 is the constant (in space) solution of $u_t = u(1-u)$. More generally, a solution which is invariant under the action of the combination of G_1 and G_2, given by $t \to t + \lambda, \quad x \to x + c\lambda$, is the *travelling wave*

$$u(x,t) = w(x-ct) \equiv w(y) \quad \text{where} \quad y = x - ct.$$

Observe that y is an invariant of the action of the group and that $w(y)$ satisfies the *ordinary* differential equation

$$-cw_y = w_{yy} + w(1-w).$$

The heat equation has many distinct similarity solutions, determined in general by additional factors such as boundary conditions, see [12]. An important such solution is the *fundamental solution*, given by

$$u(x,t) = \frac{1}{\sqrt{t}} e^{-x^2/4t} \equiv \frac{1}{\sqrt{t}} w(y)$$

where $y = x/\sqrt{t}$ is invariant under the action of the stretching group G_3 and $w(y)$ satifies the ordinary differential equation

$$w_{yy} + \frac{1}{2} y w_y + \frac{1}{2} w = 0. \tag{2.6}$$

The fundamental solution is most usually seen as the convolution kernal of a more general solution, given arbitrary initial data. It reflects all of the general solution's properties such as a decay to zero, infinite support and transport of the bulk of the solution at the rate of $\sqrt{t}$.

Finally, for the blow-up equation, there is only one self-similar solution invariant under the action of the group G_4, which is unique up to two parameters T and x_0 and which is given by

$$u(x,t) = \frac{1}{T-t} w(y), \qquad y = \frac{x - x_0}{\sqrt{T-t}}. \tag{2.7}$$

This solution becomes singular as $t \to T^-$ and $x \to x_0$. Again, $w(y)$ satisfies an ordinary differential equation

$$w_{yy} - \frac{1}{2} y w_y + w^2 - w = 0.$$

Interestingly, the function $w(y)$ gives only the beginning of a description of the true behaviour of the solution (see Section 5).

Self-similar solutions have been around for a long time and are important for the following reasons.

- Many naturally arising equations in mathematical physics possess such solutions, see the large list in [2].

- They can often be computed by solving ordinary differential equations rather than PDEs (with the self-similar solutions often appearing as homoclinic solutions of the ODE).

- They have the pleasant property that they are often *attractors* for much more general solutions with arbitrary initial data. In the dynamical system given by the partial differential equation expressed in the rescaled variables they act as attracting fixed points. Thus, nature seems to simplify itself to find solutions of partial differential equations with maximal symmetry. This means that they often describe the long-time (or the intermediate) asymptotic behaviour of the solutions. See the analytical work in this area in [25].

- They often give a good local description of the formation of a singularity or an interface (this means that the description is good away from boundaries which tend to reduce the symmetry of the equation, and at sufficiently large times so that initial conditions do not play a significant role). An example of this is given by the self-similar solution of the blow-up equation. The solution of this equation has a peak which becomes progressively higher and narrower as t tends towards the blow-up time T. Both of these features are captured in the self-similar solution which predicts a peak height of $1/(T-t)$ as $t \to T$ and a peak width proportional to $\sqrt{T-t}$.

Much effort has gone into finding automatic methods which take an arbitrary partial differential equation, find its symmetries and use these to determine similarity solutions. See for example the excellent work of Clarkson and Mansfield [9]. Whilst these methods are often effective at producing exact solutions of the equations (which can subsequently be used to test the accuracy of a numerical method) they have the disadvantage of not being able to deal with arbitrary initial and boundary conditions, of not giving information on the stability of the solution and of restricitng attention to a class of solutions (the exact self-similar solutions) wheras a more appropriate description of the evolution is that it is *approximately* self-similar. A further significant problem is that the self-similar solution itself is often determined only up to an arbitrary exponent (Barenblatt [2] refers to these as self-similar solutions of the second kind) and often the only way to find this exponent is to solve the original partial differential equation.

3 Symmetry in Numerical Methods

We now propose a class of numerical methods which makes use of the symmetry present in the partial differential equation but is not restricted so that the only solutions computable are the self-similar solutions. Potentially such methods hold great promise and have the following advantages

- If a numerical method method reflects the underlying symmetry of the partial differential equation then it will generally admit self-similar solutions as possible solutions. A well-designed method of this type should also inherit many of the stability properties of such solutions. Thus, for problems where self-similarity reflects the underlying asymptotic behaviour such methods will have good asymptotic properties.

- If symmetry governs singularity formation then these methods should perform well.

- Symmetry is often closely associated with conservation laws. If the original partial differential equation is derived from a Lagrangian then symmetries of the Lagrangian lead to conserved properties vie Noether's Theorem [20]. More generally conservation laws can be derived directly from the symmetry group [1]. The numerical method may inherit some of these conservation laws automatically.

- Symmetry can play a useful role in the error analysis of the methods.

Numerical methods for ordinary differential equations on Lie groups such as described in (2.5) which have a time stepping strategy that ensures that the solution stays on the group have become the subject of much recent analysis. This work is especially significant due to the role played by such equations in mechanics the calculation of Lyapunov exponents and in isospectral flows, and forms part of a general study of *geometric integrators*. A review of this work is given in [16]. More recently symmetry has been included in the design of methods for partial differential equations [6], [11]. Methods for including symmetry into integrators for ordinary differential equations are described in [16] and include

- Luck: various Runge Kutta methods can in certain circumstances preserve the group structure [24]

- Iterated commutator methods. These make repeated use of contructions using the commutator of the group and were originally derived by Magnus [18] with recent work by Iserles, Nørsett and Zanna [16],[17],[26].

- Tangent space and related methods. These use careful constructions to stay in the tangent bundle of the Lie group. See the work of Munthe-Kaas, Crouch and Grossmann [19], [10].

A further method which has a natural extension to partial differential equations is the use of adaptive time stepping and adaptive mesh methods. This forms the basis of our work and we concentrate on this now.

3.1 A simple ordinary differential equation example

To illustrate the effectiveness of an adaptive time stepping strategy based on symmetry when applied to an ordinary differential equation, we consider a simple example closely related to the blow-up equation. Suppose that $u(t)$ satisfies

$$\frac{du}{dt} = u^p, \quad p > 1, \quad u(0) = u_0. \tag{3.1}$$

This equation is invariant under the stretching group G_1, given by

$$t \to \lambda t, \quad u \to \lambda^{1/(1-p)} u \tag{3.2}$$

with associated differential operator

$$X_1 = t\partial_t + \frac{1}{1-p} u \partial_u.$$

It is also invariant under the symmetry group G_2 described by the diferential operator $X_2 = u^p \partial_u$. It is well known that (3.1) has the exact solution

$$u(t) = [u_0^{1-p} - (p-1)t]^{1/(1-p)} \tag{3.3}$$

for which

$$u(t) \to \infty \quad \text{as} \quad t \to T \equiv \frac{1}{p-1} u_0^{1-p}.$$

We now see how different numerical methods compare when used to compute this solution comparing (a) methods with no symmetry invariance, (b) a method which is invariant under X_1, and (c) a method which is invariant under X_2. (It is worth noting that it is rare for a numerical method to inherit *all* of the symmetries of the underlying differential equation and usually a compromise has to be sought in using a method with a reasonable subset of such symmetries.)

3.1.1 Methods with no symmetry

The simplest method to apply to (3.1) is the *forward Euler method* with a fixed step-size Δt. In this we have

$$U_{n+1} = U_n + \Delta t U_n^p, \quad t_n = n\Delta t, \quad U_n \approx u(t_n)$$

As Δt is fixed there is no possibility that this method can be invariant under any operation that involves a rescaling of time. Consequently it is not invariant under the action of X_1 (nor indeed under the action of X_2). Applying this method, we observe simply that t_n the time at the *nth*. step, increases without bound. In contrast U_n is finite for all n. Thus this scheme does not reproduce the phenomenon of finite-time blow-up.

Observe however, that the solution does become large very quickly and indeed machine overflow is reached rapidly which indicates that a singularity is forming. An analysis of this behaviour is given in [23]

If in contrast we use the implicit Euler method with a fixed time step then we arrive at the scheme

$$U_{n+1} = U_n + \Delta t U_{n+1}^p.$$

In this scheme the solution grows, until

$$U_n > \left(\frac{1}{p\Delta t}\right)^{1/(p-1)} (1 - 1/p)$$

at which point the implicit equation has no solution. This serves as a warning that something singular is occuring, but the method has again failed to reproduce the correct solution behaviour.

3.1.2 A method invariant to the action of X_1

We now construct a method which is invariant to the action of the symmetry operator X_1 which is derived from the stretching group G_1. To construct this we use a forward Euler method with an adaptive time step. Thus, setting

$$\Delta_n = t_{n+1} - t_n,$$

we use the method

$$U_{n+1} = U_n + \Delta_n U_n^p. \tag{3.4}$$

Suppose that (t_n, U_n) is a sequence of times and numerical approximations. In order to attain invariance with respect to the action of the group G_1, we require that if λ is arbitrary then $(\lambda t_n, \lambda^{1/(1-p)} U_n)$ should also be a solution. This guides us in our choice of Δ_n. Observing that under the action of the group, the quantity $u^{(p-1)}t$ is invariant, we calculate Δ_n by making $\Delta_n U_n^{p-1}$ an invariant and set

$$\Delta_n = U_n^{1-p}\Delta. \tag{3.5}$$

Here $\Delta > 0$ is a constant. The smaller the value taken by Δ the more accurate the method. The formulae (3.4),(3.5) now constitute our method and it is a simple matter to check its invariance to the action of the group.

We now proceed to calculate the accuracy of the method. Combining (3.4) and (3.5) we have simply that

$$U_{n+1} = U_n(1 + \Delta_n U_n^{p-1}) = U_n(1 + \Delta).$$

Thus

$$U_n = U_0(1 + \Delta)^n$$

and

$$t_n = \sum_{i=0}^{n-1} \Delta_i = \Delta \sum_{i=0}^{n-1} U_i^{1-p} = \Delta U_0^{1-p} \frac{(1-(1+\Delta)^{n(1-p)})}{(1-(1+\Delta)^{1-p})}.$$

Hence, as $n \to \infty$ we have that $U_n \to \infty$ and

$$t_n \to \Delta U_0^{1-p}(1-(1+\Delta)^{1-p})^{-1} = \frac{1}{p-1} U_0^{1-p} + \mathcal{O}(\Delta) = T + \mathcal{O}(\Delta).$$

Thus this method has correctly predicted that the solution will blow-up in a finite time and has estimated this time to an accuracy of $\mathcal{O}(\Delta)$, with a minimal investment of effort needed to calculate the correct time-step.

Some figures showing the performance of this method are given below. In Figure 3.1 we consider two runs of the above algorithm with $U_0 = 1$ and $U_0 = 2$ respectively and take $p = 2$ and $\Delta = 1/10$. The blow-up behaviour of both is clear. Also note that there is an exact correspondence between the solution points of the curve given by $U_0 = 1$ to the curve given by $U_0 = 2$, originating in the map $t_n \mapsto t_n/2, \quad U_n \mapsto 2U_n$.

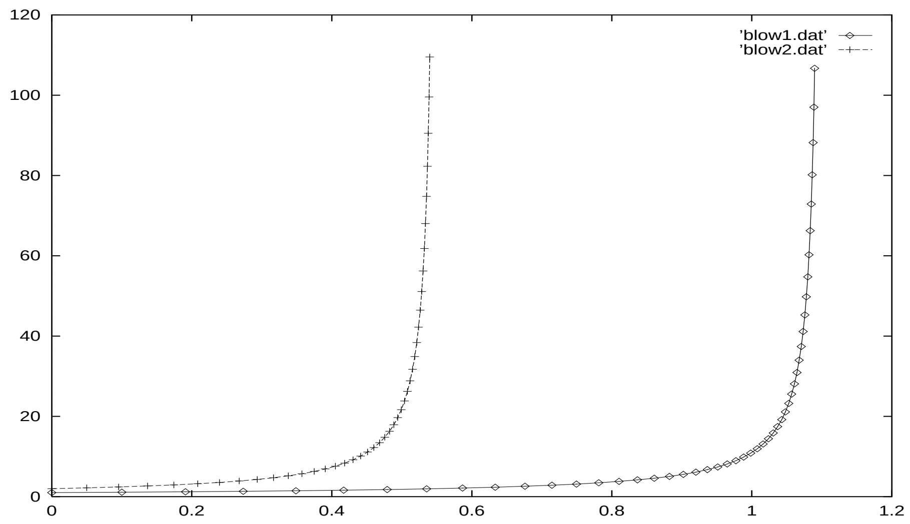

Figure 3.1: Solutions with $U_0 = 1$ and $U_0 = 2$

It is worth noting that we can analyse this method in a different way. Suppose that we have a *computational* time ζ which increases in uniform steps of size Δ and have a map $\zeta(t)$ from computational to real time so that $t_n = \zeta(n\Delta)$. Then taking the limit of $\Delta \to 0$ in (3.5) we have

$$dt/d\zeta = u^{1-p} \quad \text{and hence} \quad du/d\zeta = u.$$

Observe that this has linearised the equation which greatly simplifies the resulting analysis.

3.1.3 A method invariant under the action of the group given by X_2.

This method is somewhat harder to derive than that above. A procedure for calculating such methods is given in [11]. As there is no time transformation involved in this group action, we can use a method with a fixed time step and after some calculation we deduce that the method invariant under the action of the differential operator X_2 is given by

$$U_{n+1} = [U_n^{1-p} - (p-1)\Delta t]^{1/(1-p)}.$$

This method has both advantages and disadvantages. The advantage is that it coiincides with the solution, there is in fact no error at all. The disadvantage is that to derive it you have to do the same amount of work as it takes to analytically solve the underlying differential equation. It is thus useless as a practical numerical method.

3.2 Partial Differential Equations

Suppose now that we consider a partial differential equation in one spatial dimension. A typical example might be a reaction-diffusion equation of the general form

$$u_t = u_{xx} + f(u), \tag{3.6}$$

with associated boundary conditions. Supposing that we consider the equation alone (without reference to the boundary conditions), it is invariant under the action of a scaling group of the form

$$t \to \lambda t, \quad x \to \lambda^\alpha x, \quad u \to \lambda^\beta u.$$

To build symmetry into an associated numerical method we need to use a spatial mesh which evolves in time to allow for the rescaling of space implicit in the action of the group. Suppose for example that we consider a fully discrete finite difference method with the nth. time step given by t_n, in which u is discretised on a spatial mesh X_j^n, such that U_j^n is an approximation to $u(t_n, X_j^n)$. (Note that the mesh should not be confused with the notationally similar description of the infinitessimal generator of the group.) For such a method to be group-invariant we require that if (t^n, X_j^n, U_j^n) is a solution of the method then so is the rescaled solution $(\lambda t^n, \lambda^\alpha X_j^n, \lambda^\beta U_j^n)$. This restriction gives us a procedure for determining a suitable mesh for the problem.
To extend our earlier example, suppose that we consider the equation

$$u_t = u_{xx} + u^2,$$

invariant under

$$t \to \lambda t, \quad x \to \lambda^{1/2} x, \quad u \to u/\lambda.$$

As before, set $\Delta_n = t_{n+1} - t_n$ and, in addition, $H_j^n = X_{j+1}^n - X_j^n$. Then a symmetry-invariant way of choosing Δ_n and H_j^n is given by

$$\Delta_n = \Delta / \max_j (U_j^n) \quad \text{and} \quad H_j^n = H / \sqrt{\frac{1}{2}(U_j^n + U_{j+1}^n)}, \tag{3.7}$$

observing that both Δ_n and H_n^j are small if U_j^n is large. To solve the partial differential equation we follow [15], putting it into Lagrangian form to give

$$\frac{du}{dt} - u_x \frac{dx}{dt} = u_{xx} + u^2 \tag{3.8}$$

which can then be discretised on the (nonuniform mesh) X_j^n using (for example) a forward Euler scheme for the time stepping and a standard central difference discretisation in space. This gives the scheme

$$\frac{U_j^{n+1} - U_j^n}{\Delta_n} - \left(\frac{U_{j+1}^n - U_{j-1}^n}{X_{j+1}^n - X_{j-1}^n} \right) \left(\frac{X_j^{n+1} - X_j^n}{\Delta_n} \right) = \frac{\left(\frac{U_{j+1}^n - U_j^n}{X_{j+1}^n - X_j^n} \right) - \left(\frac{U_j^n - U_{j-1}^n}{X_j^n - X_{j-1}^n} \right)}{(X_{j+1}^n - X_{j-1}^n)/2} + (U_j^n)^2. \tag{3.9}$$

A similar exercise to that used for the related ordinary differential equation shows that the scheme (3.7),(3.9) is invariant to the action of the symmetry group, and simple experiments show that this scheme does indeed lead to solutions which become infinite in a finite time. (We compare this with a related scheme given in Section 5.)

An excellent account of such *fully invariant* discretisations and their generalisations to other problems (for example the KdV equation and Burgers' equation) is given in the work of Dorodnitsyn [11] together with a discussion of their associated conservation laws. However, they have certain disadvantages. They are difficult to set up (and solve) needing a fairly complete knowledge of the group action, they do not generally incorporate boundary conditions well, they do not often work for general initial data (indeed in many cases they assume that you are only computing self-similar solutions) and (perhaps most significantly) they are very difficult to incorporate into general-purpose software.

4 Equidistribution Methods

We now describe an alternative approach based upon the idea of *mesh equidistribution* which overcomes many of the above objections and gives invariant schemes using a simple modification of existing software. To describe these schemes we follow the ideas presented in [15].

Suppose that we consider a function $u(x,t)$ satisfying a parabolic differential equation (for example the equation (3.6) on the interval $[0,1]$. We discretise this on a moving mesh $X_j(t)$ so that $X_0(t) = 0$ and $X_N(t) = 1$ (where for the moment we treat time as continuous). The mesh can be described in terms of the mesh function $X(\xi,t)$ which maps a *computational* coordinate $\xi \in [0,1]$ into a physical coordinate such that

$$X(\frac{j}{N},t) = X_j(t).$$

To determine the location of the mesh a *monitor function* $M(x,u)$ is introduced which is typically large at points where the solution has interesting behaviour. An equidistributed mesh is then defined in terms of the identity

$$\int_0^{X(\xi,t)} M(x,u)dx = \xi \int_0^1 M(x,u)dx. \tag{4.1}$$

For computational purposes it is usual to differentiate (4.1) with respect to ξ to give

$$(MX_\xi)_\xi = 0 \tag{4.2}$$

This is referred to in [15] as *Moving Mesh PDE1* or MMPDE1. Two disadvantages of this strategy are that the resulting mesh can be unstable and also the initial mesh has to be equidistributed, which may be difficult to arrange. To overcome this problem it is usual to use a stabilised form of MMPDE1 such that the mesh tends asymptotically to the solution of (4.1) from an arbitrary initial start. A popular example of such a scheme is given by MMPDE6 in [15], whereby for a given small value of τ we have

$$X_{t\xi\xi} = -(MX_\xi)_\xi. \tag{4.3}$$

To find a solution of the partial differential equation (3.6) using the adaptive method it is first cast into Lagrangian form, giving

$$\frac{du}{dt} - u_x\frac{dx}{dt} = u_{xx} + f(u) \tag{4.4}$$

and then the full system (4.2) or (4.3) together with (4.4) is discretised using a uniform mesh in the computational coordinate ξ together with some spatial smoothing. An effective discretisation method discussed in [7] is to use a finite difference discretisation for (4.2),(4.3) and a higher order collocation method for (4.4).

Choice of the monitor function M.
There are many different ways of choosing the monitor function M. In [21] and many subsequent works M is chosen to be an estimate of the local truncation error. An alternative choice is to use the arc-length of the solution so that

$$M(x,u) = \sqrt{1+u_x^2}.$$

However neither of these reflects the underlying symmetries of the partial differential equation. A more systematic choice is *to use a monitor function M such that the complete system (4.2) or (4.3) together with (4.4), is invariant with respect to the same group of transformations as the original equation (3.6).* In this way all of the symmetry structures of th e underlying PDE will be inherited by the resulting numerical method. It is often quite easy to determine such functions and we give some examples below, (which apply to the scheme (4.3).

The blow-up equation

$$u_t = u_{xx} + u^p, \quad u(0,t) = u(1,t) = 0: \qquad M = u^{p-1}.$$

The focusing blow-up equation

$$x^q u_t = u_{xx} + u^p, \quad u(0,t) = u(1,t) = 0: \qquad M = u^{\frac{(q+2)(p-1)}{2}}.$$

The nonlinear Schrödinger equation

$$iu_t + u_{xx} + |u|^2 u = 0: \qquad M = |u|^2.$$

Given a general code to solve the system (4.2) or (4.3) together with (4.4 symmetry invariance can then be built into the method in a natural and obvious way. Observe that such a code naturally deals with arbitrary boundary and initial conditions. An example of such a general purpose code is MOVCOL, developed by Huang and Russell [14].

Time discretisation
The resulting system of ODEs that are derived from the spatial discretisation can in principle be solved by any appropriate method. For example in MOVCOL they are solved using the adaptive BDF procedures employed in the package DDASSL [22] . In practice such packages appear to be sufficiently accurate to give the correct dynamical behaviour for the solution. As an alternative, a time stepping strategy similar to that used in the last section could be used, employing an adaptive time step so that the full discretisation is group-invariant. A convenient way to think about this is to consider the (physical) time as a map $t(\zeta)$ of a computational time ζ, so that we compute on a regular mesh in the (ζ, ξ) space which is mapped to a non-uniform mesh in the (t, x) space. Of course our choice of the adptive time step Δ_n gave an explicit procedure for calculating such a function.

5 Examples

To illustrate the ideas above we now consider in more detail two examples. The first is the blow-up equation given above and the second is the porous medium equation which has a moving interface.

5.1 The blow-up equation

The blow-up equation is readily integrated using the methods described above and the given monitor function. The result of such a computation when $p = 2$ is given in Figure 5.1 in which initial data of $u(x, 0) = \cos(\pi x/2)$ leads to a solu tion which blows up at $x = \frac{1}{2}$. The sharpening nature of the blow-up peak is readily apparent. (For this problem 41 mesh points were used with a standard piecewise cubic Hermite collocation discretisation.)

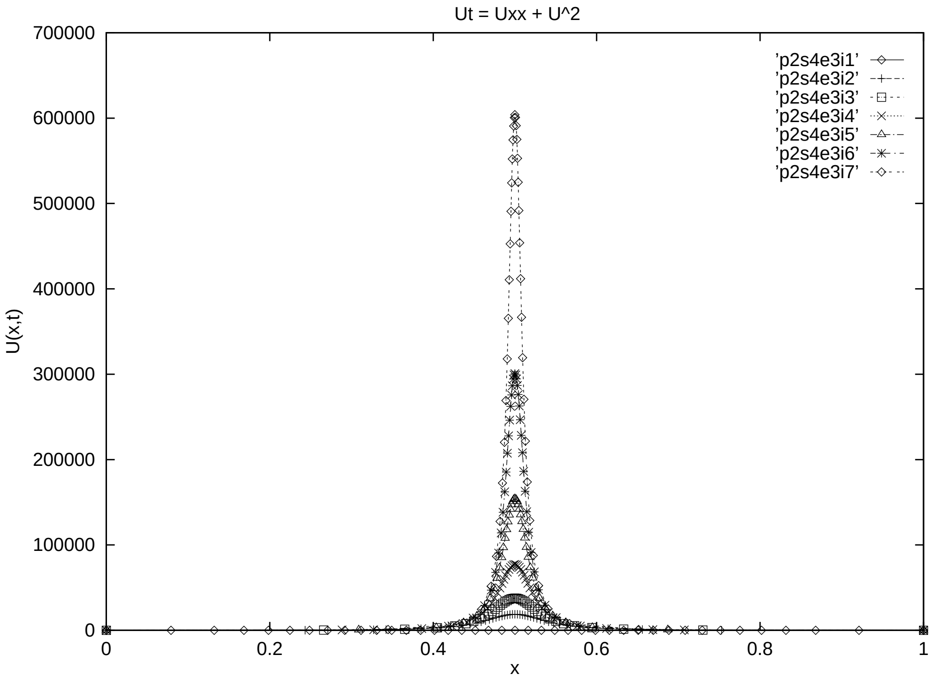

Figure 5.1: Evolution of the blow-up peak

The proposed self-similar solution to the blow-up equation takes the form

$$u(x, t) = (T - t)^{-1/(p-1)} w(y) \quad \text{where} \quad y = (x - x_0)/(T - t)^{1/2}.$$

Interestingly, when this is substituted into the partial differential equation, the only solution $w(y)$ which does not grow exponentially fast at infinity is the constant function

$$w(y) = \gamma \equiv (p - 1)^{-1/(p-1)}.$$

Detailed analysis of the partial differential equation (see for example [13]) shows that as $t \to T$

$$(T - t)^{1/(p-1)} u(x, t) \to \gamma \quad \text{on sets for which} \quad (x - x_0)/(T - t)^{1/2} \quad \text{is bounded.}$$

Thus the self-similar solution gives significant information about the solution but does not give the whole picture. Indeed, a numerical method which were to simply solve

the ordinary differential equation resulting from the transformation would give a mis leading answer (as was indeed the case in early computations on blow-up.)
The correct dynamics is obtained by introducting a slow time $s = \log(T-t)$ and studying the dynamics of the function $w(y,s)$ given by

$$u(x,t) = (T-t)^{-1/(p-1)} w(y,s) \quad \text{where} \quad y = (x-x_0)/(T-t)^{1/2}.$$

On any compact set in y, $w(y,s) \to \gamma$ as $s \to \infty$. The evolution in the neighbourhood of γ can be studied using centre manifold theory. The resulting structure of $u(x,t)$ is then given by the *approximately self-similar* solution $w(y,s)$ which evolves toward the true self-similar solution. Indeed it can be shown that $w(y,s)$ can be expressed as a function of a new variable $\eta = y/s$, which becomes a natural coordinate for the problem. Observe here how the scaling group plays an essential role in determining the dynamics, even though the final solution is not quite group invariant.

This behaviour is reflected very well in the numerical method. Due to the scaling invariance of the method, it admits solutions which can be expressed in the form

$$U_i(t) = (T-t)^{-1/(p-1)} W_i \quad \text{and} \quad X_i(t) = x_0 + (T-t)^{1/2} Y_i.$$

Indeed, W_i and Y_i constant are both possible solutions, but these are not admitted as W_i would also grow exponentially fast as $i \to \infty$. Instead the method automatically converges to a solution in which

$$X_i(t) = x_0 + (T-t)^{1/2} \log(T-t), \quad U_i(t) = U_i(t) = (T-t)^{-1/(p-1)} W_i$$

accurately reflecting the correct approximate self-similar behaviour of the solution. More details are given in [6]. This calculation has demonstrated the flexibility of the group invariant method. It is interesting to compare this with other methods which implement scaling directly for example [3], which achieve similar results, but which uses purpose-designed methods with very many more mesh points.

5.2 The porous medium equation

This equation which describes (amongst other phenomena) the flow of a gas through a porous medium is given by

$$u_t = (uu_x)_x = \frac{1}{2}(u^2)_{xx}, \quad |u| \to 0 \quad \text{as} \quad |x| \to \infty, \quad u(x,0) = u_0(x) \geq 0. \tag{5.1}$$

Observe that the diffusion of u is zero if $u = 0$. As a result, if $u_0(x)$ has its support in the interval $[s_-, s_+]$ so that $u_0(s_-) = u_0(s_+) = 0$, then in its subsequent evolution the function $u(x,t)$ has compact support with sharply defined interfaces $s_-(t)$ and $s_+(t)$. It can be shown that

$$\dot{s}_- = -u_x(s_-, t), \quad \dot{s}_+ = -u_x(s_+, t). \tag{5.2}$$

and that the fisrt integral and first moment of u are constant so that

$$\int_{-\infty}^{\infty} u dx = \text{const}, \quad \int_{-\infty}^{\infty} x u dx = \text{const}. \tag{5.3}$$

The porous medium equation is invariant under translations in time, space, reflexions in x and the action of the scaling group

$$t \to \lambda t, \quad x \to \lambda^{\alpha} x, \quad u \to \lambda^{\beta} u \quad \text{where} \quad 2\alpha - \beta = 1.$$

As a result it admits self-similar solutions of the form

$$u(x,t) = (t-t_0)^{\alpha} w(y) \quad \text{where} \quad y = (x-x_0)/(t-t_0)^{\beta}. \tag{5.4}$$

For this solution to satisfy the conservation laws (5.3) we must have $\alpha = 1/3, \quad \beta = -1/3$. Unlike the previous problem, it can be proven that the self-similar solution is an attractor in the rescaled coordinate system [25] and thus gives the correct long time asymptotics of the system.

A numerical method to solve this problem should have the following properties: it should correctly compute the interfaces, it should have good conservation properties and it should have an attracting self-similar solution. All of these three des irable properties are shared by group-invariant methods. To make things transparent in this case, we will use a simple finite difference method rather than the collocation method described earler. To implement this we place mesh points $X_i \quad i = 0 \d{o} tsN$ in the support of u and set $U_0 = 0, \quad U_N = 0$ and $U_i > 0, \quad i = 1...N-1$. A simple discretisation of the Lagrangian form of (5.1) for $i = 1 \ldots N-1$ is then given by

$$\dot{U}_i - \left(\frac{U_{i+1} - U_{i-1}}{X_{i+1} - X_{i-1}}\right)\dot{X}_i = \frac{\left(\frac{U_{i+1}^2 - U_i^2}{X_{i+1} - X_i}\right) - \left(\frac{U_i^2 - U_{i-1}^2}{X_i - X_{i-1}}\right)}{X_{i+1} - X_{i-1}}. \tag{5.5}$$

For $i = 0$ or $i = N$ we discretise (5.2) to give

$$\dot{X}_0 = -U_1/(X_1 - X_0), \quad \dot{X}_N = U_{N-1}/(X_N - X_{N-1}). \tag{5.6}$$

It is then easy to see that (5.5),(5.6) is invariant with respect to the actions of all the symmetry groups acting on the original equation. A further calculation reported in [8] shows that U_i satisfies the conservation law

$$\sum_{i=0}^{N-1} (X_{i+1} - X_i)(U_{i+1} + U_i) = \text{const}. \tag{5.7}$$

The evolution of this equation is fairly gentle and it is appropriate in this case to use MMPDE1 given in (4.2). A simple monitor function leading to group invariance is $M = 1$ resulting in the invariant equation

$$X_{i+1} - 2X_i + X_{i-1} = 0 \tag{5.8}$$

for the mesh. The system (5.5,5.6,5.8) can then be integrated.

We observe immediately that the discrete problem admits a discrete self-similar solution of the form

$$U_i(t) = t^{\alpha} W_i, \quad X_i = t^{\beta} Y_i \tag{5.9}$$

and applying (5.7) we deduce immediately that $\alpha = -\frac{1}{3}, \beta = \frac{1}{3}$. Thus the discrete system has the correct long-time dynamics. Furthermore, it can be shown using matched asymptotic expansions [8] that

$$\max_i |W_i - w(Y_i)| \sim C \log(N)/N^2 \quad \text{as} \quad N \to \infty.$$

Thus the discrete self similar solution converges to the true self-simlar solution under mesh refinement. Furthemore, if we set $s = \log(t)$ and $W_i(s) = t^{1/3} U_i(t) \quad Y_i(s) = t^{-1/3} X_i(t)$ then an analysis of a rescaling of the system (5.5),(5.6),(5.8) about the self-similar solution shows that this latter solution is linearly stable. Numerical experiments indicate strongly that it is also a global attractor. In Figure 5.2 we show the resulting solution, plotting both the unscaled variables X_i, U_i and the scaled variables W_i, Y_i.

The numerical method is thus performing optimally. For large times it converges to the discrete self-similar solution, which in turn converges to the true self-similar solution under mesh refinement. Further details are given in [8]

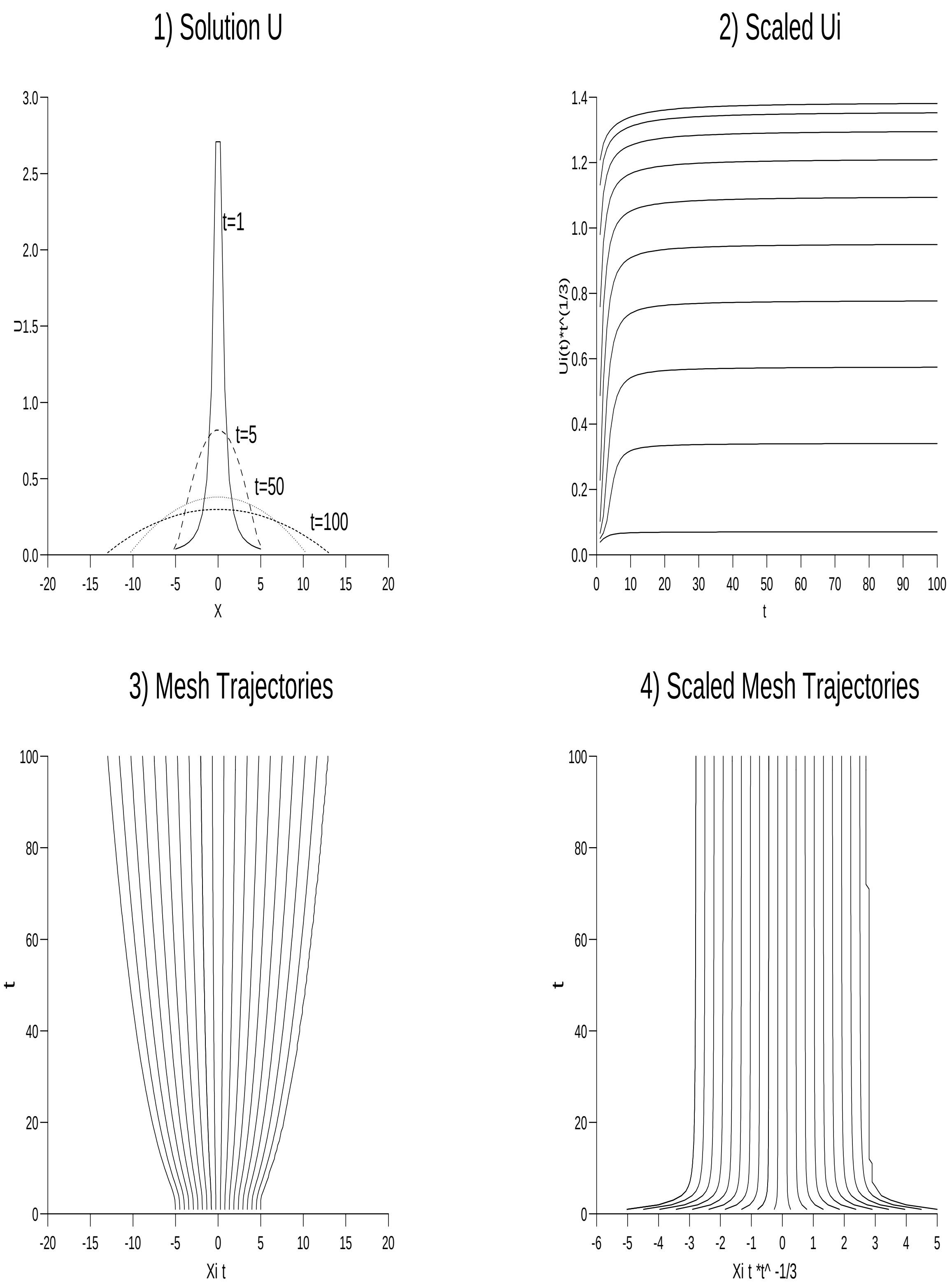

Figure 5.2: Porous medium equation $u_t = (uu_x)_x$ solved on a mesh with $N = 20$.

6 Conclusions

We have demonstrated that symmetry plays an important role in studies of partial differential equations, numerical methods which are symmetry invariant perform well in practice and that, using equidistribution methods, they can be implemented relatively easily using existing software. They hold much promise for effectiveness on a wide class of symmetry invariant PDEs such as the nonlinear Schrödinger equation, the Ginsberg Landau equation, the turbulent burst equation, Fisher's equation and many more examples from mathematical physics. Some significant questions still remain to be answered before these methods can be used as a general tool. For example, how well do they adapt to higher dimensions, how do they cope with problems with ill defined symmetries (such as the self-similar solutions of the second kind described in [2]), how do they deal with problems for which the action of the symmetry group is very local (such as in the formation of a front) or only applies asymptotically? All these are excellent topics for further research.

References

[1] S. Anco and G. Bluman, "Direct construction of conservation laws from field equations", Phys. Rev. Lett., **78**, (1997), pp. 2869–2873.

[2] G.I. Barenblatt, "Self-similarity and intermediate asymptotics", (1996), CUP

[3] M. Berger and R.V.Kohn, "A rescaling algorithm for the numerical calculation of blowing-up solutions." Comm. Pure Appl. Math. (1988), pp. 841–863.

[4] G. Bluman and J. Cole, "Similarity methods for differential equations", (1974), Springer-Verlag.

[5] G. Bluman and S. Kumei, "Symmetries and differential equations", (1989), Springer, New York.

[6] C.J. Budd, W. Huang and R.D. Russell, "Moving mesh methods for problems with blow-up", SIAM J. Sci. Comp. **17**, (1996), pp. 305–327.

[7] C.J. Budd, J. Chen, W. Huang and R.D. Russell, "Moving mesh methods with applications to blow-up problems for PDEs", in "Numerical Analysis 1995", ed. D. Griffiths and G. Watson, Pitman Research Notes in Mathematics Series **344**, (1996), pp. 1–18, Longman,

[8] C.J. Budd and G.J. Collins, "An invariant moving mesh scheme for the nonlinear diffusion equation", (1997), to appear in Appl. Numer. Math.

[9] P. Clarkson and E. Mansfield, "Symmetry reductions and exact solutions of a class of nonlinear heat equations", Physica D, **70**, (1993), pp. 250–288.

[10] P.E. Crouch and R. Grossman, "Numerical integration of ordinary differential equations on manifolds", J. Nonlinear Sci., **3**, (1993), pp. 1–33.

[11] V. Dorodnitsyn, "Finite difference methods entirely inheriting the symmetry of the original equations", in *Modern Group Analysis: Advanced analytical and computational methods in mathematical physics*, ed. N. Ibragimov, (1993), pp. 191-201.

[12] L. Dresner, "Similarity solutions of nonlinear partial differential equations", Pitman Research Notes in Mathematics, **88**, (1983), Longman.

[13] V. Galaktionov and S. Posashkov, "The equation $u_t = u_{xx} + u^\beta$. Localisation and asymptotic behaviour of unbounded solutions", Differentsial'nye Uravneniya, **22**, (1986), pp. 1165–1173.

[14] W. Huang and R.D.Russell, "A moving collocation method for time dependent PDEs" Appl. Numer. Math., **20**, (1996), pp. 101–116.

[15] W. Huang, Y. Ren and R.D.Russell, "Moving mesh methods based on moving mesh partial differential equations", J. Comp. Phys. **112**, (1994), pp. 279–290.

[16] A. Iserles, "Ordinary differential equations on (and off) manifolds" manifolds." in *Foundations of Computational Mathematics* (F. Cucker, ed.), (1997), Springer-Verlag, New York.

[17] A. Iserles and S. Nørsett, "Linear systems in Lie groups", Proceedings of IMACS World Congress, Berlin, (1997).

[18] W. Magnus, "On the exponential solution of differential equations for a linear operator", Comm. Pure and Appl. Math. **7**, (1954), pp. 649–673.

[19] H. Munthe-Kaas, "Lie-Butcher theory for Runge-Kutta methods", BIT **35**, (1995), pp. 572–587.

[20] P. Olver, "Applications of Lie groups to differential equations", (1986), Springer, New York.

[21] V. Pereyra and E. G. Sewell, "Mesh selection for discrete solution of boundary value problems in ordinary differential equations", Numer. Math. **23**, (1975), pp. 261–268.

[22] L.R. Petzold, "A description of DASSL: A differential/ algebraic system solver", Sandia Ntational Labs. Report **SAND82-8637**, (1982) Amer. Math. Soc. **17**, 301-325 (1916)

[23] J.M. Sanz-Serna and J.G. Verwer, "A study of the recursion $y_{n+1} = y_n + \tau y_n^m$", J. Math. Anal. and Appl. **116**, (1986), pp. 456-464.

[24] J.M. Sanz-Serna and M. Calvo, "Numerical Hamiltonian problems", (1994), Chapman and Hall, London.

[25] J.L Vazquez, "An introduction to the mathematical theory of the porous medium equation", in *Shape optimisation and free boundaries* M.Delfour ed. (1992), pp. 347–389, Kluwer Academic.

[26] A. Zanna and H. Munthe-Kaas, "Iterated commutators, Lie's reduction method and ordinary differential equations on matrix Lie groups", in *Foundations of Computational Mathematics* (F. Cucker & M. Shub, eds.), (1997), pp. 434–441, Springer-Verlag, New York.

Acknowledgement

This research was supported by an earmarked grant from the EPSRC. We would like to thank Arieh Iserles for helpful comments on an earlier draft of this paper.

Chris Budd
Dept. of Mathematical Sciences
University of Bath
Bath, BA2 7AY
UK
Gordon Collins
School of Mathematics
University Walk
Bristol, BS8 1TW
UK