A comparison of three methods for solving one-dimensional time-dependent immersed interface problems

Nikolas M. Skoufis
Supervisor: Michael A. Page

Date of submission: October 25, 2013

Contents

Introduction

The Immersed Interface Method (IIM) is a numerical technique for solving differential equations where either the solution, derivatives of the solution or parameters in the equation have discontinuities over the domain. It is fundamentally a sharp interface method, meaning that it attempts to preserve the sharpness of discontinuities instead of smearing them as with other methods such as Peskin's Immersed Boundary Method. Since its inception it has proven to be a robust method that is easily modified and adapted to solve a wide variety of problems in the physical and natural sciences. It has also been extensively improved, adapted and combined with existing methods to solve problems that would be otherwise difficult to solve due to the inability of standard methods to deal with discontinuities.

This thesis will begin with an overview of the immersed interface method including some applications, before discussing and comparing three different methods for solving time dependent PDEs in one spatial dimension.

Part I

Background on the immersed interface method

1 Introduction to the immersed interface method

The IIM stemmed from a need to accommodate discontinuities in solutions or coefficients more generally than with the previously popular Immersed Boundary Method [14]. The Immersed Boundary Method was developed initially in [31] and [32] as an extension to the Marker and Cell computational fluid dynamics technique [33]. The MAC method was capable of simulating fluid flow around rigid boundaries by dividing the domain into a rectangular mesh, however it could not accommodate moving boundaries that deformed during the simulation of the flow. The method employed by Viecelli was popularized by Peskin in his papers modelling the flow of blood in the heart ([25] and [26]) and named the Immersed Boundary Method (IB). The IB method models the boundaries as fluid particles which move with the local velocity of the fluid, but exert a force on the surrounding fluid. This force at the boundary is modeled as a smoothed Dirac delta function [27], but this leads to the interface being 'smeared' up to the grid size. Although the immersed boundary method is still used in many fields, (particularly in Peskin's field of bio-mechanics eg. [9], [37]) the IIM has some advantages over the immersed boundary method.

The IIM was first developed and outlined in a thesis by Li in 1994 [15] and a paper based on this work by Leveque and Li [14]. These papers develop the IIM for general one and two dimensional elliptic problems and heat equation problems, with both stationary and moving interfaces. A key advantage of the IIM is that it exactly enforces discontinuities in parameters or the solution and its derivatives across boundaries, where as Peskin's method smooths these discontinuities in order to model them. The smoothing is accomplished by introducing a discrete delta function that approximates any singularities. However the accuracy of this method is limited by the size of the grid. In contrast to the IB method, the IIM attempts to correct the solution only at the discontinuity, and does so in a way which is independent of the grid. The result of this is that the IIM accurately preserves sharp boundaries where as the IB method does not. This can be seen clearly in figure 1.

Another advantage of the IIM is that it allows for discontinuities in any parameter, as well as discontinuities that must be enforced in the solution and its derivatives. This allows for a much wider variety in the problems that can be solved using the IIM.

This introductory part will be organized into four sections. The first section will briefly introduce the IIM in 1D and provide an example of how the IIM works. The second section will discuss the problems and solutions associated with applying the IIM to parabolic, elliptic and hyperbolic PDEs. The third section will discuss some augmentations and modifications to the IIM. The fourth section will provide some novel examples of how the IIM has been used to solve a range of problems.

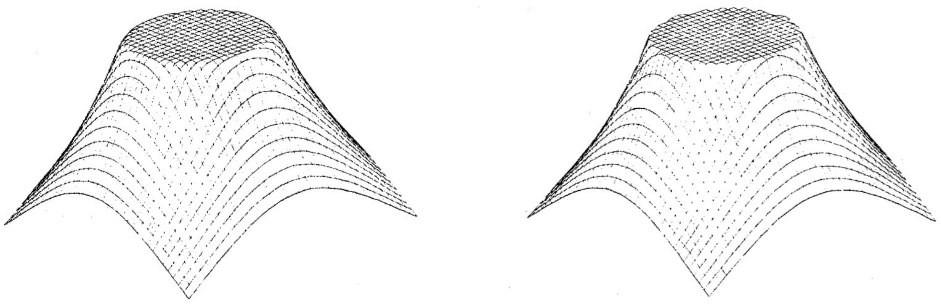


Figure 1: Solution to an elliptic interface problem using (a) a discrete delta function such as in Peskin's method and (b) the IIM. Note that discontinuity is much sharper when using the IIM. Figure from [14]

2 The IIM for a simple 1D interface problem

The basic outline for solving a problem using the IIM in 1D is to:

- Determine jump conditions
- Discretise the equation using a particular finite difference method
- Separate domain into regular and irregular points
- Set up and solve a system that minimizes the truncation error at the irregular points
- Solve the modified finite difference equations

As an illustration, we will solve a second order linear constant coefficient ODE

$$a\frac{d^2u}{dx^2} + b\frac{du}{dx} + cu = 0, \tag{1}$$

with $0 \leq x \leq 1$, a, b and c arbitrary constants, and $u(0) = u_0$, $u(1) = u_1$.

For this problem we will specify particular jump conditions, but in general the jump conditions might be determined from the physical processes that causes the discontinuities in a problem.

A jump condition defines the relationship between the function (and its derivatives) on either sides of the discontinuity (referred to as the interface). If the discontinuity occurs at $x = \alpha$ then we represent the jump condition for $u(x)$ by $[u]_\alpha$ where

$$[u]_\alpha := u^+(\alpha) - u^-(\alpha), \tag{2}$$

with

$$\begin{aligned} u^+(\alpha) &:= \lim_{x\to\alpha^+} u(x), \\ u^-(\alpha) &:= \lim_{x\to\alpha^-} u(x), \end{aligned}$$

and similarly for $[\frac{du}{dx}]_\alpha$ and $[\frac{d^2u}{dx^2}]_\alpha$. These conditions will be used to correct the finite difference equations near the interface to preserve the accuracy of the finite difference method.

We can set up a grid of x values in the usual way

$$x_i = \Delta x i, \qquad i = 0, 1, 2, \ldots, N$$

for some number of points N with

$$\Delta x = \frac{1}{N}.$$

Note that the IIM works equally well regardless of whether the interface at $x = \alpha$ lies on a gridpoint. This feature is particularly useful when considering problems where the interface is moving, as it means creating a new mesh at each timestep is not necessary.

In order to maintain accuracy with the least amount of extra computation and complication, the finite difference scheme remains unmodified for points that lie away from the interface. The points not immediately adjacent to the interface are referred to as **regular** points, while the two points (or one point if the interface lies on a grid-point) closest to the interface are referred to as **irregular** points. If we take

$$x_j, x_{j+1} : x_j \leq \alpha \leq x_{j+1}$$

to be the irregular points, then

$$x_i : i \neq j, j+1 \tag{3}$$

are the regular points. If we approximate the value of $u(x_i)$ by u_i we can obtain the familiar three-point centered difference scheme for the regular points

$$a\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + b\frac{u_{i+1} - u_{i-1}}{2h} + cu_i = 0. \tag{4}$$

If we collect terms based on u values this becomes

$$u_{i+1}\left(\frac{a}{h^2} + \frac{b}{2h}\right) + u_i\left(\frac{-2a}{h^2} + c\right) + u_{i-1}\left(\frac{a}{h^2} - \frac{b}{2h}\right) = 0. \tag{5}$$

For irregular points, we must generate finite difference coefficients so as to minimize the truncation error at the interface. The general form of the finite difference equation is

$$\gamma_{l,1}u_{l-1} + \gamma_{l,2}u_l + \gamma_{l,3}u_{l+1} = C_l, \qquad l = j, j+1, \tag{6}$$

with $\gamma_{l,1}$,$\gamma_{l,2}$,$\gamma_{l,3}$,C_l constants to be determined.

We determine the $\gamma_{l,m}$ so as to ensure that the truncation error T_l at the point $x = \alpha$ matches the order of the finite difference method used.

The truncation error is given by

$$T_l = \gamma_{l,1}u(x_{l-1}) + \gamma_{l,2}u(x_l) + \gamma_{l,3}u(x_{l+1}) - C_l - a\frac{d^2u}{dx^2} - b\frac{du}{dx} - cu \tag{7}$$

for $l = j, j+1$.

In order to evaluate this expression, it is necessary to take Taylor series expansions of each of the approximate terms and expand them around the point $x = \alpha$. This gives the following expressions:

$$\begin{aligned}
u(x_{j-1}) &= u^-(\alpha) + (x_{j-1} - \alpha)\frac{du}{dx}^-(\alpha) + \frac{1}{2}(x_{j-1} - \alpha)^2\frac{d^2u}{dx^2}^-(\alpha) + O(\Delta x^3),\\
u(x_j) &= u^-(\alpha) + (x_j - \alpha)\frac{du}{dx}^-(\alpha) + \frac{1}{2}(x_j - \alpha)^2\frac{d^2u}{dx^2}^-(\alpha) + O(\Delta x^3),\\
u(x_{j+1}) &= u^+(\alpha) + (x_{j+1} - \alpha)\frac{du}{dx}^+(\alpha) + \frac{1}{2}(x_{j+1} - \alpha)^2\frac{d^2u}{dx^2}^+(\alpha) + O(\Delta x^3).
\end{aligned} \tag{8}$$

But using our jump conditions we can express $u(x_{j+1})$ strictly in terms from the left hand side of the interface. Substituting this equation back into the equation for the truncation error and collecting like terms we can obtain and solve a linear system of equations for γ that ensures the desired accuracy at $x = \alpha$.

$$\begin{cases} \gamma_{j,1} + \gamma_{j,2} + \gamma_{j,3} = c \\ \gamma_{j,1}(x_{j-1} - \alpha) + \gamma_{j,2}(x_j - \alpha) + \gamma_{j,3}(x_{j+1} - \alpha) = b \\ \frac{\gamma_{j,1}}{2}(x_{j-1} - \alpha)^2 + \frac{\gamma_{j,2}}{2}(x_j - \alpha)^2 + \frac{\gamma_{j,3}}{2}(x_{j+1} - \alpha)^2 = a. \end{cases} \tag{9}$$

We also require that

$$C_j = \gamma_{j,3}\left([u]_\alpha + (x_{j+1} - \alpha)\left[\frac{du}{dx}\right]_\alpha + \frac{1}{2}(x_{j+1} - \alpha)^2\left[\frac{d^2u}{dx^2}\right]_\alpha\right) \tag{10}$$

to give $T_j = O(\Delta x^3)$.

The derivation for the difference equation at $x = x_{j+1}$ is identical except now there are two points that fall to the right of the interface. Following the same process we see that the only difference is the C_{j+1} term which is now

$$\begin{aligned} C_{j+1} = \gamma_{j,2}&\left([u]_\alpha + (x_{j+1} - \alpha)\left[\frac{du}{dx}\right]_\alpha + \frac{1}{2}(x_{j+1} - \alpha)^2\left[\frac{d^2u}{dx^2}\right]_\alpha\right) \\ + \gamma_{j,3}&\left([u]_\alpha + (x_{j+2} - \alpha)\left[\frac{du}{dx}\right]_\alpha + \frac{1}{2}(x_{j+2} - \alpha)^2\left[\frac{d^2u}{dx^2}\right]_\alpha\right). \end{aligned} \tag{11}$$

We now have a complete system of equations which when solved produces a second order accurate solution to our original problem.

Although here we have considered a fairly simple problem as an illustration, the IIM can be completely generalized for non-constant coefficients that may include discontinuities and for higher orders with little difficulty. In addition, the IIM is easily implemented for higher order finite difference schemes [19]. However, in order to keep problems simple, this project only considers first and second order finite difference schemes.

3 The IIM for elliptic, parabolic and hyperbolic interface problems

3.1 Elliptic problems

Much of the analysis in section 2 extends to elliptic problems in 2D such as Poisson's equation

$$\nabla^2 u = f. \tag{12}$$

However there are some differences when generalizing to two and higher dimensions.

- Irregular points are now defined to be any points whose finite difference stencil is crossed by the interface.
- A point (x_i^*, y_i^*) must be chosen chosen on the interface that is nearby the irregular point about which to expand the Taylor series.
- A coordinate transformation into coordinates normal and tangential to (x_i^*, y_i^*) simplifies the interface relations.
- In addition to the normal finite difference stencil, an additional point is often required to ensure that the system for γ is non-singular. In [8] the authors present a method for choosing the additional point if necessary based on ensuring the diagonal dominance of the resulting system for γ.

Jump conditions are then derived based on these new coordinates. In 2D, these jump conditions usually take the form of

$$\begin{aligned} [u]_\Gamma &= w(s), \qquad (13)\\ \left[\frac{\partial u}{\partial n}\right]_\Gamma &= v(s), \end{aligned}$$

where Γ is a closed curve in the domain that defines the position of the interface, n is the unit normal, s is the arc length parameter along the interface Γ and $v(s)$ and $w(s)$ define the jump conditions which we assume to be specified or otherwise determined from the physical properties of the problem.

Once these have been determined, a system of equations can be generated to solve for the unknown finite difference coefficients at each irregular point. These coefficients can then be inserted into the larger system and the interface problem can be solved.

Some added complexity is inherent when solving elliptic equations due to the requirement of solving a large linear system accurately and quickly. If the parameters in the elliptic equation are piece-wise constant, then a fast method exists for solving the system [16]. This method involves solving a modified system which is easier to solve, and has the useful property that the number of iterations required for convergence is not related to the mesh size or the size of the jump in the parameters.

If the parameters of the equation are not piece-wise constant, the previously mentioned method for selecting the sixth point by Fogelson and Keener [8] proves challenging to integrate with existing linear solvers such as the multigrid method. However Li and Ito [18] have developed a method that generates a diagonally dominant coefficient matrix with symmetric negative definite part. These properties allow it to be used with the modified multigrid method as shown by Adams and Li [1].

Although these approaches allow fast linear solvers to be used, they do not attack the primary problem which is that the system generated by the IIM is not symmetric and is not positive definite. The work of Berthelson [4] and more recently Zhao et. al. [41] aims to solve this by using standard finite difference coefficients at irregular points and using the correction term only to compensate for the discontinuity. This approach is promising, and has shown improved accuracy over the original formulation of the IIM and other modifications to the IIM.

These techniques have been extended to three dimensions eg. [6], [7].

3.2 Parabolic problems

The IIM can also be used to solve parabolic problems such as the 2D heat equation

$$\frac{\partial u}{\partial t} = \kappa \nabla^2 u. \tag{14}$$

Similar issues that were discussed in terms of elliptic problems in section 3.1 occur when solving 2D parabolic problems. However there is added complexity in that since there is a time dependence there may now be a discontinuity in $\left[\frac{\partial u}{\partial t}\right]$ over the interface and this must be corrected for when choosing our γ values and correction terms. This is achieved using a similar process of Taylor series expansion as found in section 2 along with derivatives of the boundary condition with respect to t.

An inherent problem with parabolic problems that is not present when solving elliptic problems is that unconditionally stable implicit methods require the solution of extremely large implicit systems in two and higher dimensions. One solution to this are so called 'operator splitting' methods which split each time-step into two half steps with one spatial coordinate implicit and one explicit per time-step. A popular splitting method is the ADI method, and this has been adapted for use with the IIM in Li [15] and Li and Mayo [21]. This was achieved by splitting the correction term C_j into x and y components in order to match the splitting of operators employed in the ADI method.

Extra complexity associated with time dependent problems is that the interface may be moving with respect to the domain. These problems are relatively easy to solve when the position of the interface is prescribed, but become significantly more difficult when the position of the interface must be solved for along with solving the parabolic system, such as for the 'Stefan problem' of a melting crystal.

There are a number of methods that may be employed to find the position of the boundary or interface. The commonality between all of these methods is that they first use the IIM to solve the equations governing the motion of the interface and then employ some sort of evolution scheme to evolve the position of the boundary between time-steps.

The simplest of these methods is the front tracking method. This method employs a number of control points which define the position of the boundary and are evolved at each time step according to the local properties of the system (eg. velocity). A more complex method is the level set method, which uses the zero level set of a level set function to track the interface. The time derivative of this function can be coupled with the local velocity and then the position of the boundary can be solved for. Level set methods are generally more robust than front tracking methods because they are able to model processes where there is merging or splitting of boundaries. The level set method has been used in conjunction with the IIM to solve the aforementioned Stefan problem, as well as the Hele-Shaw flow [19].

3.3 Hyperbolic problems

The IIM can be used to solve hyperbolic problems such as the 2D wave equation

$$\frac{\partial^2 u}{\partial t^2} = c^2 \nabla^2 u. \tag{15}$$

The IIM for hyperbolic equations has been developed by Zhang [39] and Zhang and Leveque [40] relatively early in the history of the IIM. The techniques from section 2 are again valid for use in higher dimensional hyperbolic systems, however there are some modifications due to the necessity of different discretisation schemes. Zhang and Leveque [40] employ both the second order Lax-Wendroff scheme and more accurate wave propagation algorithms provided by the CLAWPACK library [13], although the latter is based on a finite volume discretisation.

In contrast to elliptic and parabolic PDEs, little work has been done on the IIM for hyperbolic PDEs. A possible explanation for this is that the point of the IIM is to only differ from standard finite difference methods where necessary ie. at interfaces. Thus, the two possible complications with modifying finite difference schemes at interfaces (namely movement of the interface and interfaces in 2D and higher dimensional problems) have already been dealt with when developing the IIM for parabolic and elliptic equations.

4 Augmentations and modifications of the IIM

4.1 The Explicit-Jump Immersed Interface Method (EJIIM)

The Explicit-Jump Immersed Interface Method (herein EJIIM) has been develop by Wiegmann [34] and Wiegmann and Bube [35], [36].

Instead of generating new finite difference coefficients (the γs found above), the EJIIM derives correction terms for the existing finite difference methods using Taylor series and the known jumps over the interfaces. This is advantageous as it avoids having to solve a linear system for the γs and it also allows for higher order finite difference schemes to be generated by creating higher order correction terms. However this method can only be employed if the jump conditions are known explicitly.

Recently, Rutka [29] and Rutka and Li [30] have used the EJIIM to solve the Navier-Stokes equations for two fluids with a singular force term at the boundary. The EJIIM is particularly suited to this problem as standard augmented methods (which are required due to a coupling between jump conditions for the pressure and velocity over the interface) are difficult to implement due to the challenge in finding augmented variables and equations.

4.2 The Matched Interface and Boundary method (MIB)

The Matched Interface and Boundary (herein MIB) method has been developed in [42], [43] and [44]. The MIB method treats points either side of the interface as being in separate domains, introducing fictitious points where necessary to complete the finite difference stencil. These fictitious points are then coupled with their counterparts on the other side of the interface and the jump conditions. This allows for the fictitious points to be solved for and then the finite difference stencils can be completed. Since we now have accurate estimates on the fictitious points, we can repeat this algorithm to generate further points and thus increase the accuracy of our final solution by using a larger finite difference stencil. This is a great advantage over the standard IIM because it implies that increased accuracy can be obtained via an algorithm instead of via manual computation as with the IIM.

In practice, Zhou et. al. [44] have achieved 16th order accuracy when the interfaces are straight lines, and 6th order accuracy for elliptic problems with curved interfaces.

4.3 Immersed Finite Element Methods (IFEMs)

The IIM using a finite element (FE) formulation has been developed in [17], [20] and [11]. For standard FE methods, second order accuracy is guaranteed if the interface lies on a grid-point. However in higher dimensions it may be difficult or costly to ensure this, especially when the interface is moving. The IFEM attempts to retain this accuracy while allowing for a uniform grid that does not have to conform to interfaces.

5 Applications of the IIM

The IIM has been applied to many varied problems, both linear and non-linear. Some particular examples will be given here, but for a more complete list consult Li and Ito [19].

5.1 The IIM for electro-migration voiding problems

Electro-migration voiding problems deal with the movement and deformation of non-conducting voids in thin, metal, current carrying lines such as circuit traces found on the surfaces of printed circuit boards. This group of problems is part of a larger group of problems dealing with constant volume diffusion along surfaces under the influence of external forces. For an overview of this class of problems, see [5].

The solution of electro-migration voiding problems using the IIM is considered in [23]. Electro-migration voiding problems involve solving a fourth order nonlinear PDE on a rectangular domain. The IIM is superior to other methods for solving this type of problem for two reasons. It is less computationally intensive than other methods, and the grid can remain fixed between time steps as opposed to being recalculated after every time step in other methods. The authors develop a numerical scheme based on the IIM which allows them to numerically model some instances of void migration as show in Figure 2.

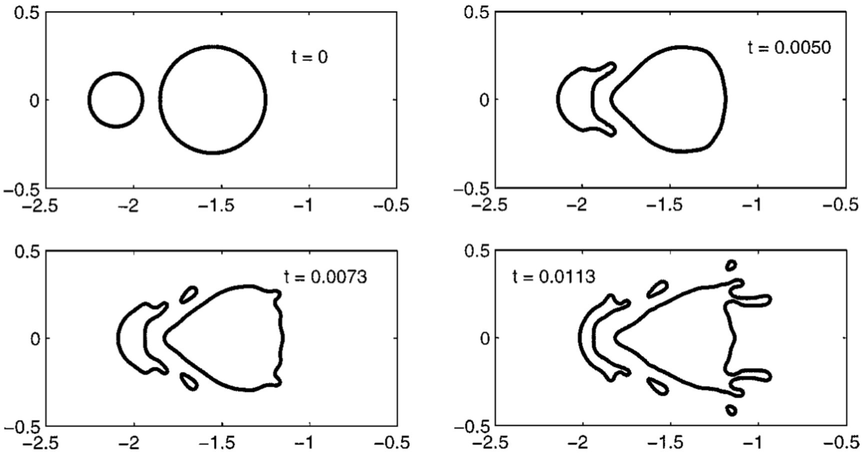


Figure 2: The simulated evolution of two voids over time. Reprinted from [23].

5.2 The IIM for electro-capacitance tomography problems

The IIM has been applied to problems in the field of electro-capacitance tomography (ECT) by considering the IIM in polar coordinates by Alvarez et. al. [2]. The usual setup of an ECT machine is a cylindrical tube with some material passing through it. Around the edges of the tube there are between 8 and 16 electrodes. Pairs of electrodes are chosen, and the capacitance between them is measured. This process of choosing pairs of electrodes is repeated until all electrode combinations have been exhausted.

There are two problems of interest in ECT. The 'forwards problem' is to calculate the capacitance that would be measured by electrodes placed around the edges of the ECT machine give a know electromagnetic permittivity inside the device. The 'inverse problem' is to reconstruct the electromagnetic permittivity of the space inside the ECT instrument from a series of capacitance reading by electrodes located around the edge of the device. ECT is a relatively immature analytical tool, but it has been applied in a few areas including flame monitoring and monitoring of fluidised beds in laboratory tests, and analysis of bead milling and pneumatic conveying processes in industrial tests ([38] and references therein). ECT is particularly useful in monitoring mixtures of different materials, because if these materials possess different permittivities, the spatial location of the interfaces between them will be clearly visible in imaging data obtained from ECT machines.

The IIM is a useful tool for solving problems in ECT because the permittivity will have a jump across the interface between different media that are flowing through the ECT device. Alvarez et. al. [2] perform a coordinate transformation from cartesian to cylindrical coordinates and set up a numerical scheme for solving the elliptical equation that arises in ECT problems. The numerical scheme is shown to provide second order convergence against exact results, and numerical experiments are conducted to solve the forwards problem for some given permittivity distributions (see Figure 3).

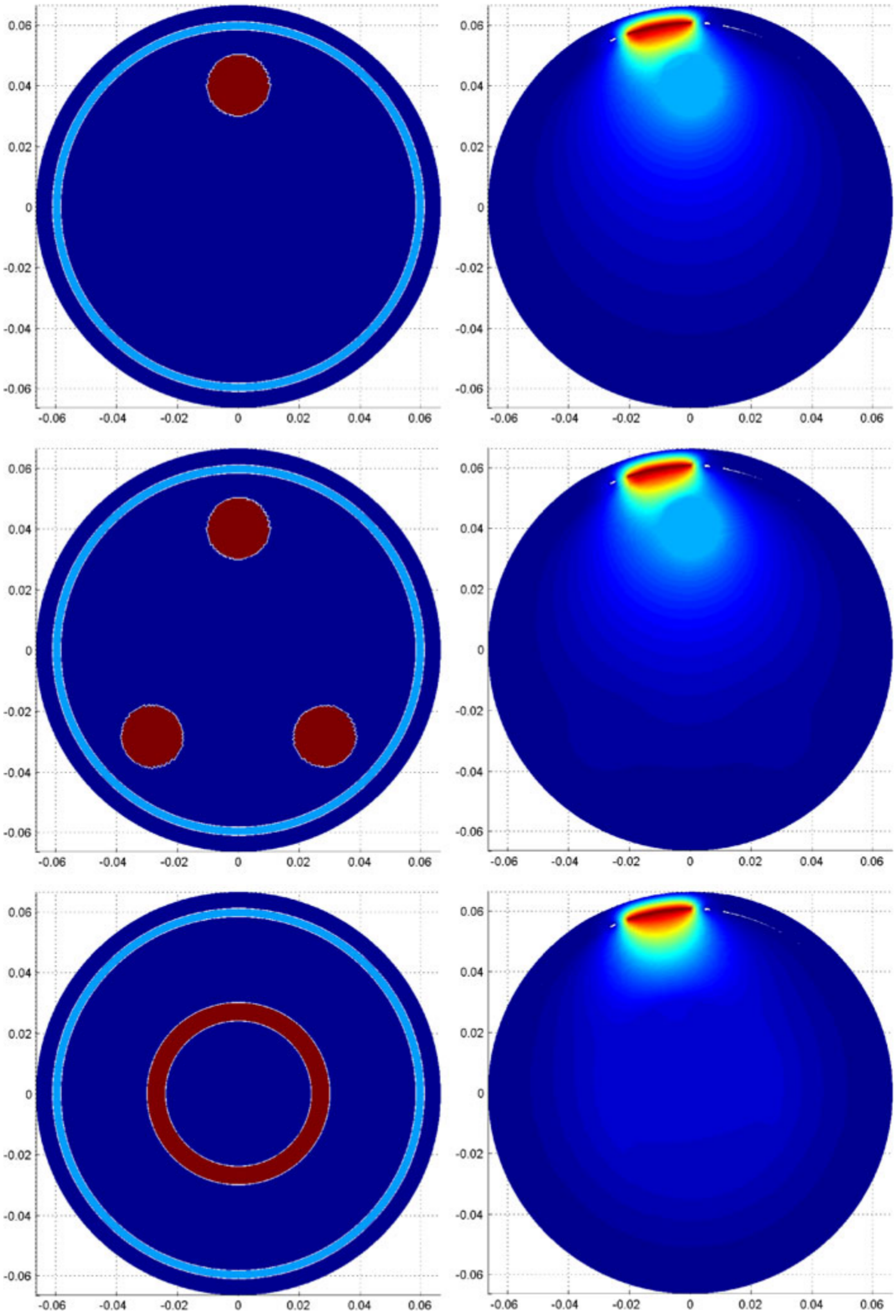


Figure 3: Solutions (right) calculated from given permittivities (left). Reprinted from [2].

5.3 The IIM for traffic flow problems

The IIM has been used to model problems in traffic flow on highways by Wiegmann and Bube [35]. In this paper, the authors develop a model for traffic flow based on some simple assumptions about driver behaviour and conservation of the number of cars. Discontinuities can arrise in traffic flow problems either due to initial conditions (eg. cars starting from traffic lights) or due to shocks. The authors begin by confirming that the IIM still retains accuracy when it is used to solve non-linear DEs before modelling a non-linear equations for traffic flow. An example of the results obtained from this analysis can been seen in Figure 4.

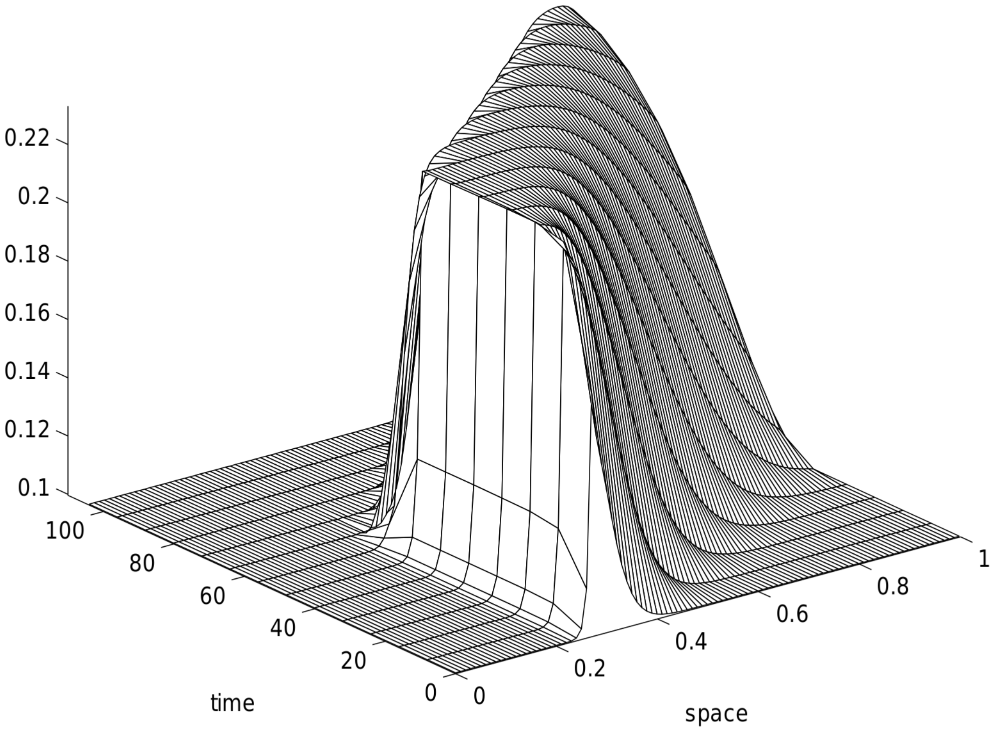
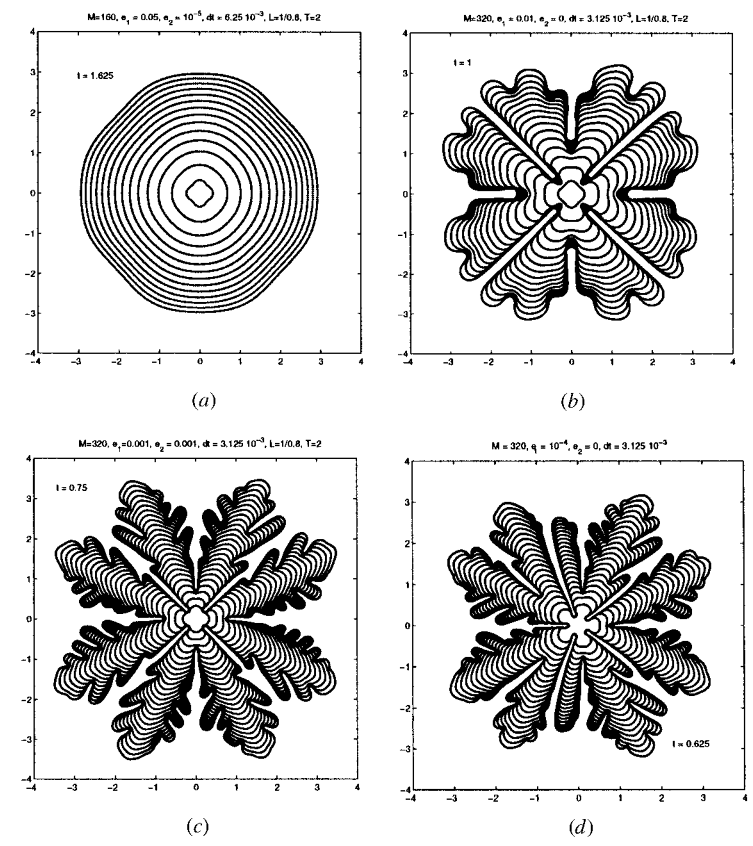


Figure 4: A simulation of a constant inbound flow of cars followed by an abrupt stop. The vertical axis here is car density. Note the wave-like propagation of the influx of cars, as well as the diffusion of density away from the congested areas. Reprinted from [35].

5.4 The IIM for crystal growth

The complex patterns of snowflakes are produced by an interplay between heat diffusion, surface tension and a moving interface. This makes it an ideal problem to solve using the IIM. In general, crystal growth problems occur any time a compound is slowly changing phases eg. solid to liquid in the case of a snowflake. This process involves heat diffusing through the different phases at different rates, as well as heat being generated by the phase change at the boundary and a curvature and surface tension dependant condition at the boundary. For a more thorough treatment of the subject of crystal growth consult [12]. Li and Soni [22] use the IIM with both a modified Crank-Nicholson scheme and alternating directional implicit (ADI) method to solve problems of crystal growth. Some numerical results can be seen in Figure 5.



(a) (b)

(c) (d)

Figure 5: Crystal growth with different values for surface tension. The figures are labelled in order of decreasing surface tension. It is evident that lower values of surface tension encourage greater dendrite growth. Reprinted from [22].

5.5 The IIM for insect flight

Bergou et. al. [3] have use the IIM in conjunction with other methods to study the phenomenon of wing pitch reversal in insect flight.

Wing pitch reversal is a very rapid adjustment of the angle of the wing relative to the direction of airflow (angle of attack) just prior to the upstroke of the insects wing motion. In [3] the authors use the IIM to solve the Navier-Stokes equations governing airflow around the wing by modelling the wing surface as a singular force. This allows them to calculate the aerodynamic components of the forces applied to the wing during wing pitch reversal. From the data generated by the IIM and other studies of insect flight, the authors conclude that wing pitch reversal in insects is in fact passive. That is, no action is required by the insect to cause wing pitch reversal as it is merely a consequence of the aerodynamic force exerted on the wing by the surrounding air.

Figure 6 shows a snapshot of the vorticity field as calculated by the IIM, just prior to the rotational motion of wing pitch reversal.

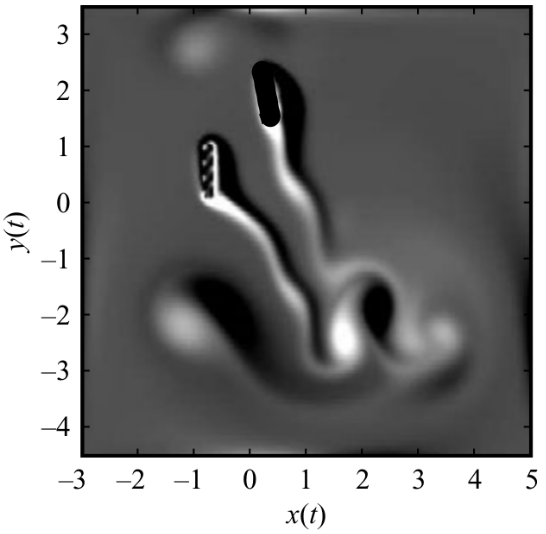


Figure 6: A snapshot of vorticity just prior to rotation of the wing, calculated using the IIM. The dark oval areas in the upper left are the two wings on one side of a dragonfly. Reprinted from [3].

Part II

Three methods for solving time-dependent PDEs

6 Introduction

This chapter aims to compare three different methods that can be used to solve one-dimensional time-dependent PDEs. All of these methods are variants of the IIM with small adjustments made to the way they calculate the solution.

The first method to be considered is proposed by Li and Ito [19]. When solving problems that are time dependent, and extra correction is required to account for the possibility of a discontinuity in $\partial u/\partial t$ at the interface. Li and Ito's method uses Taylor series expansion to derive a constant term which is added to the difference equation.

The second method is a new method that I am proposing called the undetermined time coefficient method. This method attempts to reduce error at the interface by solving for the finite difference coefficients that approximate $\partial u/\partial t$ in the difference equation in such a way as to minimize error at the interface. This modification is justified in an identical fashion to the modifications made to finite difference coefficients in the one-dimensional time-independent IIM discussed in section 2

The third method due to Russell and Wang [28] takes a different approach. The Russell and Wang method corrects errors near the interface purely by adding a constant forcing term to the difference equation, instead of modifying the finite difference coefficients as with the other two methods.

The description of these three methods will be followed by some comments on how difficulties can arise when more complex discretisation schemes are employed than the simple spatially second order schemes employed in this project, or when multiple spatial dimensions are required.

The next section will outline and solve some simple test problems with known solutions in order to assess the accuracy of the different methods

The final section will consist of concluding remarks and some possible avenues for further research.

7 The Li and Ito method

The first method we will consider is presented in Li and Ito [19]. We will illustrate this method by demonstrating its derivation using the heat equation and a forward time centered space differencing scheme, but it is equally applicable to other equations and differencing schemes.

We wish to solve the equation

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2} \tag{16}$$

on $0 \leq x \leq 1$, $t > 0$ and with given initial and boundary conditions

$$u(x,0) = u_0(x), \qquad u(0,t) = a, \qquad u(1,t) = b, \tag{17}$$

with jump conditions

$$[u]_\alpha := u^+(\alpha) - u^-(\alpha) = w(t), \qquad \left[\frac{\partial u}{\partial x}\right]_\alpha := \frac{\partial u}{\partial x}^+(\alpha) - \frac{\partial u}{\partial x}^-(\alpha) = v(t) \tag{18}$$

at some moving interface

$$x = \alpha(t). \tag{19}$$

We can discretize this by choosing small stepsizes Δx and Δt and letting

$$x_i = i\Delta x, \quad i = 0, 1, \ldots, N \quad x_N = 1 \tag{20}$$

$$t_k = k\Delta t, \quad k = 0, 1, \ldots \tag{21}$$

Now we approximate the differential equation using forward differences in time and centered differences in space. As with the IIM for ODEs, we will use a standard discretisation scheme at regular points and a modified scheme near the interface. We will use the FTCS method in these overviews of the interface correction methods to illustrate the point.

The difference equation we obtain for regular points is

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} = \kappa \frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{(\Delta x)^2} \tag{22}$$

where u_i^k is an approximation to $u(x_i, t_k)$.

This gives that

$$u_i^{k+1} = u_i^k + \frac{\kappa \Delta t}{\Delta x^2}\left(u_{i-1}^k - 2u_i^k + u_{i+1}^k\right) \tag{23}$$

$$\Rightarrow \quad u_i^{k+1} = u_{i-1}^k \frac{\kappa \Delta t}{\Delta x^2} + u_i^k \left(1 - 2\frac{\kappa \Delta t}{\Delta x^2}\right) + u_{i+1}^k \frac{\kappa \Delta t}{\Delta x^2} \tag{24}$$

At irregular points, we can use the standard IIM method to discretise $\partial^2 u/\partial x^2$ but we require a different method to ensure that the discretisation for $\partial u/\partial t$ is second order accurate. Reduction of accuracy can occur when the interface crosses the grid line $x = x_i$ at some time $t = \tau$, $t_k < \tau < t_{k+1}$. This may result in a discontinuity in $\partial u/\partial t$ which means that

$$\lim_{t \to \tau^+} \frac{\partial u}{\partial t}(t) \neq \lim_{t \to \tau^-} \frac{\partial u}{\partial t}(t) \tag{25}$$

In order to correct for this we add a term to the approximation to $\partial u/\partial t$ which is derived in a similar way to the derivation of the IIM in section 2.

We begin by expanding $u(x_i, t_{k+1})$ and $u(x_i, t_k)$ about the grid crossing time $t = \tau$.

$$u(x_i, t_k) = u^-(x_i, \tau) + (t_k - \tau)\frac{\partial u}{\partial t}^-(x_i, \tau) + \mathcal{O}(\Delta t^2) \tag{26}$$

$$u(x_i, t_{k+1}) = u^+(x_i, \tau) + (t_{k+1} - \tau)\frac{\partial u}{\partial t}^+(x_i, \tau) + \mathcal{O}(\Delta t^2) \tag{27}$$

Using the jump conditions we can write equation 27 as

$$u(x_i, t_{k+1}) = u^-(x_i, \tau) + [u]_\tau + (t_{k+1} - \tau)\frac{\partial u}{\partial t}^-(x_i, \tau) + (t_{k+1} - \tau)\left[\frac{\partial u}{\partial t}\right]_\tau + \mathcal{O}(\Delta t^2) \tag{28}$$

Combining these two expressions gives

$$u(x_i, t_{k+1}) - u(x_i, t_k) = \Delta t \frac{\partial u}{\partial t}^-(x_i, \tau) + [u]_\tau + (t_{k+1} - \tau)\left[\frac{\partial u}{\partial t}\right]_\tau + \mathcal{O}(\Delta t^2) \tag{29}$$

Hence,

$$\frac{\partial u}{\partial t}^-(x_i, \tau) = \frac{u(x_i, t_{k+1}) - u(x_i, t_k)}{\Delta t} - \frac{[u]_\tau}{\Delta t} - \frac{(t_{k+1} - \tau)}{\Delta t}\left[\frac{\partial u}{\partial t}\right]_\tau + \mathcal{O}(\Delta t) \tag{30}$$

We can determine $[u]_\tau$ from the jump conditions by setting $\alpha = \alpha(\tau)$, but $[\partial u/\partial t]_\tau$ requires some derivation.

$$u^+(\alpha, t) - u^-(\alpha, t) = w(t) \tag{31}$$

Differentiating with respect to t gives

$$\left(\frac{\partial u}{\partial x}^+(\alpha, t) - \frac{\partial u}{\partial x}^-(\alpha, t)\right)\frac{d\alpha}{dt} + \frac{\partial u}{\partial t}^+(\alpha, t) - \frac{\partial u}{\partial t}^-(\alpha, t) = w'(t) \tag{32}$$

Hence

$$\left[\frac{\partial u}{\partial t}\right] = w'(t) - \left[\frac{\partial u}{\partial x}\right]\frac{d\alpha}{dt} \tag{33}$$

We can now solve the system using standard techniques.

8 The unknown time coefficient method

The second technique we will consider involves using a similar system of undetermined coefficients as in the one-dimensional IIM.

Solving an identical problem as in section 7, we obtain the difference equation for the FTCS method at points away from the interface

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} = \kappa \frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{(\Delta x)^2} \tag{34}$$

However at points near the interface, we wish to modify this equation in order to retain accuracy at the discontinuity. We do this by replacing the coefficients of u_i^{k+1} and u_i^k with coefficients which will be determined with the goal of minimizing truncation error at the interface.

Let

$$\frac{\partial u}{\partial t}(x_i, t_k) = \gamma_{k,1} u_i^k + \gamma_{k,2} u_i^{k+1} + C_k \tag{35}$$

Expanding the values of u on the right hand side as Taylor series about the grid crossing time $t = \tau$ gives

$$u(x_i, t_k) = u^-(x_i, \tau) + (t_k - \tau)\frac{\partial u}{\partial t}^-(x_i, \tau) + \mathcal{O}(\Delta t^2) \tag{36}$$

$$u(x_i, t_{k+1}) = u^+(x_i, \tau) + (t_{k+1} - \tau)\frac{\partial u}{\partial t}^+(x_i, \tau) + \mathcal{O}(\Delta t^2) \tag{37}$$

Since

$$u^+(x_i, \tau) = u^-(x_i, \tau) + [u]_\tau \tag{38}$$

$$\frac{\partial u}{\partial t}^+(x_i, \tau) = \frac{\partial u}{\partial t}^-(x_i, \tau) + \left[\frac{\partial u}{\partial t}\right]_\tau \tag{39}$$

We can rewrite equation 37 as

$$u(x_i, t_{k+1}) = u^-(x_i, \tau) + [u]_\tau + (t_{k+1} - \tau)\left(\frac{\partial u}{\partial t}^-(x_i, \tau) + \left[\frac{\partial u}{\partial t}\right]_\tau\right) + \mathcal{O}(\Delta t^2) \tag{40}$$

Substituting this back into the equation for $\partial u/\partial t$ gives that

$$\frac{\partial u}{\partial t}(x_i, t_k) = \gamma_{k,1} \left\{ u^-(x_i, \tau) + (t_k - \tau) \frac{\partial u}{\partial t}^-(x_i, \tau) \right\} \tag{41}$$

$$+ \gamma_{k,2} \left\{ u^-(x_i, \tau) + [u]_\tau + (t_{k+1} - \tau) \left(\frac{\partial u}{\partial t}^-(x_i, \tau) + \left[\frac{\partial u}{\partial t} \right]_\tau \right) \right\} + C_k + \mathcal{O}(\Delta t^2) \tag{42}$$

By collecting coefficients of γ we can see that if

$$\begin{cases} \gamma_{k,1} + \gamma_{k,2} = 0 \\ \gamma_{k,1}(t_k - \tau) + \gamma_{k,2}(t_{k+1} - \tau) = 1 \end{cases} \tag{43}$$

and

$$C_k = -\gamma_{k,2} \left([u]_\tau + (t_{k+1} - \tau) \left[\frac{\partial u}{\partial t} \right]_\tau \right) \tag{44}$$

then

$$\gamma_{k,1} u_i^k + \gamma_{k,2} u_i^{k+1} + C_k = \frac{\partial u}{\partial t}(x_i, t_k) + \mathcal{O}(\Delta t^2). \tag{45}$$

We can now solve the system of equations using standard techniques.

9 The Russell and Wang method

The final method is due to Russell and Wang [28]. This method does not alter the coefficients of the difference equation at all, but instead adds constant correction terms based off of the jump conditions.

As in the previous two sections, we are solving the heat equation with the conditions given in section 7 using the FTCS method.

As in those sections, our difference equation at points away from the interface is

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} = \kappa \frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{(\Delta x)^2} \tag{46}$$

However this difference equation is not valid for points near the interface.

In order to compute the values near the interface u_j^k and u_{j+1}^k, observe that if $\tilde{u}_{j+1}^{k}$ is the value obtained by solving the system using values from the left of the interface as if there were no interface and corresponding jumps then

$$u_{j+1}^k = \tilde{u}_{j+1}^{k} - [u] - (x_{j+1} - \alpha)\left[\frac{\partial u}{\partial x}\right] - \frac{1}{2}(x_{j+1} - \alpha)^2 \left[\frac{\partial^2 u}{\partial x^2}\right] + \dots \tag{47}$$

and similarly for the point at x_j. Hence, we can effectively solve the system by adding the extra terms in the above expression to the RHS of the difference equation.

Similarly, to correct for discontinuities in time we can use an identical derivation to find that

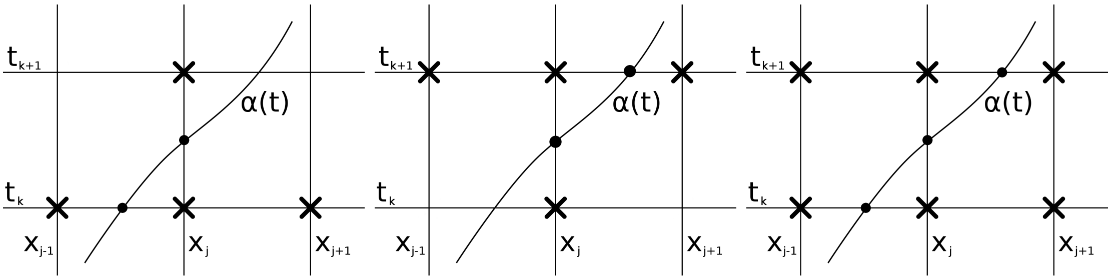
$$u_j^{k+1} = \tilde{u}_j^{k+1} - \operatorname{sgn}\left(\frac{d\alpha}{dt}\right)\left\{[u] + (t_{k+1} - \tau)\left[\frac{\partial u}{\partial t}\right] + \frac{1}{2}(t_{k+1} - \tau)^2\left[\frac{\partial^2 u}{\partial t^2}\right] + \dots\right\} \tag{48}$$

and similarly for u_j^k.

10 Adapting the IIM methods to other discretisation schemes

This section discusses the adaptations necessary to use the methods mentioned in the previous three sections with more complex discretisation schemes than the FTCS method.

The addition of extra points to the finite difference stencil increases the complexity of the IIM because there are now extra locations where the interface crosses the grid, and thus extra locations that require correction. Figures 7a, 7b and 7c illustrate this with three spatially second-order discretisations with different time discretisations.



(a) Forward-time (b) Backward-time (c) Crank-Nicolson

Figure 7: The finite difference stencil for three different spatially second order discretisation schemes. The interface $\alpha(t)$ is drawn, and the locations where the interface cuts the finite difference stencil are indicated with black dots. These locations will require a time correction if they lie on an x gridline or a space correction if they lie on a t gridline

In addition to varying with the discretisation scheme used, the location of these correction points also varies with the sign of $d\alpha/dt$. Although the examples considered in this project all use monotonic functions for $\alpha(t)$, there may be situations where the movement of the interface displays some nonmonotonic behavior. It is also necessary to consider what effect this would have on problems in two spatial dimensions, where the grid crossings can now occur in two dimensions, as well as in the time dimension if the interface is moving (see figure 8).

The addition of extra interface-grid crossings requires extra corrections. Although these corrections are not complex to derive based on any of the three above methods, their implementation is where difficulties arise. Properly accounting for every possible combination of grid crossings is a difficult task to accomplish programattically, and this is one disadvantage of the IIM not shared by smoothing methods.

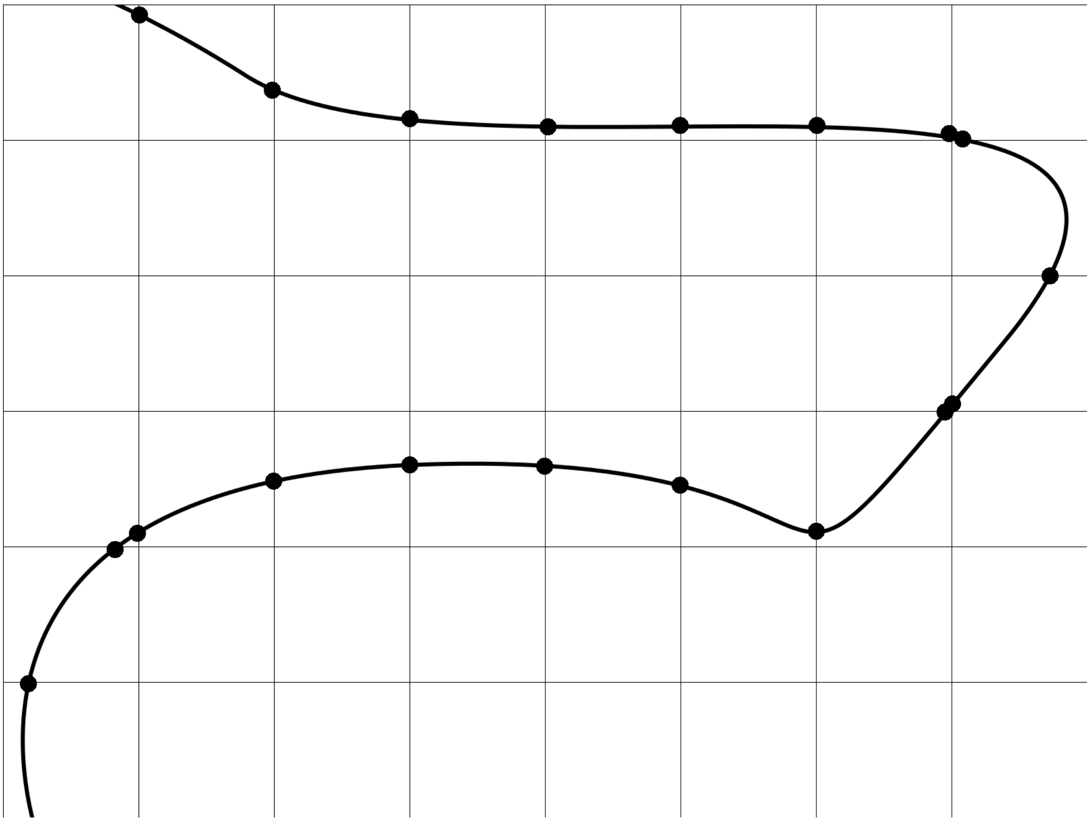


Figure 8: This diagram shows the effect of an extra dimension on the number of grid crossings. In a 2D problem, corrections would need to be made at every point where there is a dot. If the interface were moving, this would potentially have to be redone at every timestep, in addition to doing corrections in the time dimension.

11 The IIM for the 1D heat equation with moving interface

In this section we solve the heat equation with a moving interface using the IIM to modify some simple finite difference schemes.

We are solving the heat equation

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2} \tag{49}$$

with $u = u(x,t)$ on $x \in [0,1]$, $t > 0$ and $\kappa > 0$, and with initial and boundary conditions

$$u(x,0) = U_0(x), \qquad u(0,t) = A, \qquad u(1,t) = B, \tag{50}$$

and $\alpha(t) \in [0,1]$ the location of the interface.

We also have jump conditions

$$[u]_\alpha = u^+(\alpha) - u^-(\alpha), \tag{51}$$

$$\left[\frac{\partial u}{\partial x}\right]_\alpha = \frac{\partial u}{\partial x}^+(\alpha) - \frac{\partial u}{\partial x}^-(\alpha), \tag{52}$$

$$\left[\frac{\partial^2 u}{\partial x^2}\right]_\alpha = \frac{\partial^2 u}{\partial x^2}^+(\alpha) - \frac{\partial^2 u}{\partial x^2}^-(\alpha). \tag{53}$$

11.1 Solving for initial condition of two error functions

The problem we consider is that where the initial condition is

$$U_0(x) = \begin{cases} \mathrm{Erfc}(\frac{x}{\sqrt{t_0}}) & : x \le \alpha(t_0) \\ 2\,\mathrm{Erfc}(\frac{1-x}{\sqrt{t_0}}) & : x > \alpha(t_0) \end{cases} \tag{54}$$

where $\mathrm{Erfc}(z)$ is the complementary error function and with $\kappa = \frac{1}{4}$, $u(0,t) = 1$, $u(1,t) = \frac{1}{2}$ and $\alpha(t)$ is the function which satisfies

$$\mathrm{Erfc}\left(\frac{\alpha(t)}{\sqrt{t}}\right) = 2\,\mathrm{Erfc}\left(\frac{1-\alpha(t)}{\sqrt{t}}\right) \tag{55}$$

which must be determined numerically.

The exact solution to this problem is

$$u(x,t) = \begin{cases} \mathrm{Erfc}(\frac{x}{\sqrt{t}}) & : x \le \alpha(t) \\ 2\,\mathrm{Erfc}(\frac{1-x}{\sqrt{t}}) & : x > \alpha(t) \end{cases} \tag{56}$$

We wish to solve this problem approximately using the various methods mentioned in the previous section combined with various discretisation schemes.

For all the schemes, let

$$x_i = i\Delta x, i = 0, 1, 2, \ldots, N \qquad t_k = k\Delta t, k = 0, 1, 2, \ldots \tag{57}$$

and

$$u_i^k \approx u(x_i, t_k) \tag{58}$$

We now consider a simple example of the use of the IIM to solve a problem where the exact solution is known. Although the problem is somewhat contrived, knowledge of the solution allows us to evaluation the accuracy of the approximate solutions.

11.1.1 The forward difference method

The forward difference method is an explicit method accurate to $\mathcal{O}(\Delta x^2)$ in space and $\mathcal{O}(\Delta t)$ in time, with

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} = \kappa \frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{\Delta x^2} \tag{59}$$

at points away from the interface. At points adjacent to the interface we modify the finite difference schemes using the Li and Ito method, the undetermined time coefficient method and the Russell and Wang method.

All three methods perform fairly well. Their errors are of the expected magnitude and are stable over the same range of Δt and Δx values as for continuous problems.

The only anomalous results is that both the Li and Ito method and the undetermined time coefficient methods do not show monotonic error behavior. Instead, their error seems to oscillate with no obvious pattern, although it does tend to increase leading up to an irregular timestep before being corrected once the time correction occurs.

11.1.2 The backwards difference method

The backwards difference method is an implicit method accurate to $\mathcal{O}(\Delta x^2)$ in space and $\mathcal{O}(\Delta t)$ in time, with

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} = \kappa \frac{u_{i-1}^{k+1} - 2u_i^{k+1} + u_{i+1}^{k+1}}{\Delta x^2} \tag{60}$$

at points away from the interface. At points adjacent to the interface we modify the finite difference scheme using the three methods.

Again, all methods perform well. The errors are of the correct magnitude, but the oscillations that were present for the forward difference method are still present for the Li and Ito and undetermined time coefficient methods.

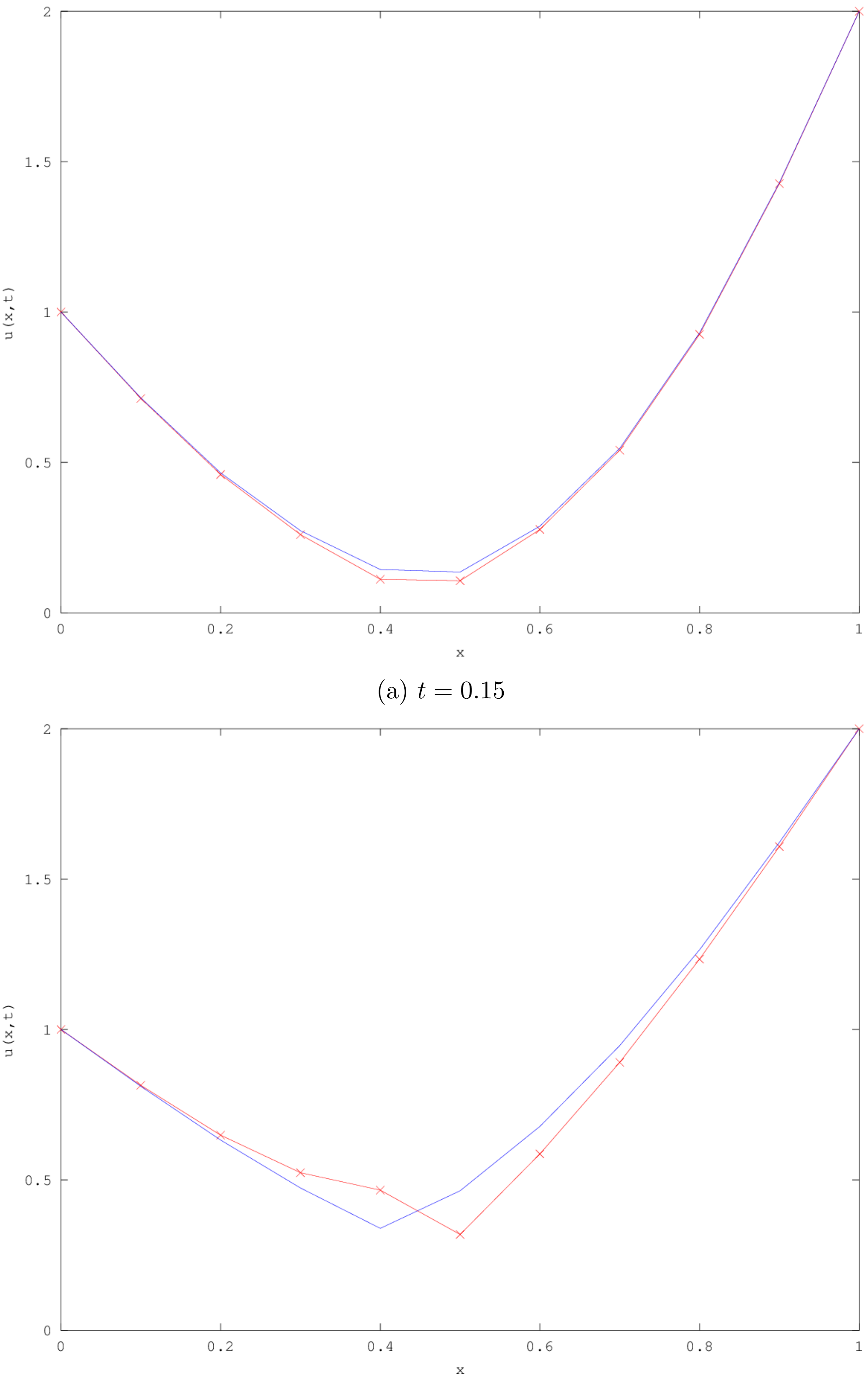
11.1.3 The Crank-Nicolson method

The Crank-Nicolson method is an implicit method which is accurate to $\mathcal{O}(\Delta x^2)$ in space and $\mathcal{O}(\Delta t^2)$ in time, with

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} = \frac{\kappa}{2}\left(\frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{\Delta x^2} + \frac{u_{i-1}^{k+1} - 2u_i^{k+1} + u_{i+1}^{k+1}}{\Delta x^2}\right) \tag{61}$$

For this method, the Russell and Wang method behaves with the correct error magnitude, however the other two methods do not.

Both the Li and Ito and undetermined time coefficient methods develop large errors near the interface, indicating that they are not correcting sufficiently for the discontinuity as shown in figure 9



(a) $t = 0.15$

(b) $t = 0.35$

Figure 9: Solutions for the Li and Ito method with Crank-Nicolson discretisation for small and large values of t. The solid line represents the exact solution while the line with × symbols represents the approximate solution. $\Delta x = 0.1$, $\Delta t = 0.01$, $t_0 = 0.1$.

12 The IIM for the advection equation

In this section, we will explore the IIM for solving the advection equation in one dimension with a non-smooth initial condition

12.1 The IIM for a wedge initial condition

We will solve the advection equation

$$\frac{\partial u}{\partial t} = -a\frac{\partial u}{\partial x} \tag{62}$$

on $0 \leq x \leq 1$ using a number of methods.

The parameters of this problem are:

interface at $x = \alpha(t)$ where

$$\alpha(t) = 0.5 + at \tag{63}$$

initial conditions

$$u(x,0) = \begin{cases} 1 - x & x < \alpha(0) \\ x & x \geq \alpha(0) \end{cases} \tag{64}$$

and jump conditions that match the discontinuities in the initial condition

$$[u]_\alpha = 0$$
$$\left[\frac{\partial u}{\partial x}\right]_\alpha = 2$$

We also have the inflow boundary condition that $u(0,t) = 1$.

This has exact solution

$$u(x,t) = \begin{cases} 1 + at - x & x < \alpha(t) \\ x - at & x \geq \alpha(t) \end{cases} \tag{65}$$

12.1.1 FTCS method

The forward time centered space method is $\mathcal{O}(\Delta x^2)$ in space and $\mathcal{O}(\Delta t)$ in time, and is unconditionally unstable.

The FTCS method is a first order method derived by taking a centered difference approximation in space and a forward difference in time

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} = -a \frac{u_{i+1}^k - u_{i-1}^k}{2\Delta x} \tag{66}$$

or

$$u_i^{k+1} = u_i^k - \frac{a\Delta t}{\Delta x}(u_{i+k}^n - u_{i-1}^k) \tag{67}$$

As in the previous section, this difference equation is valid for points away from the interface, but near the interface we must correct using one of our three techniques.

For the advection equation, flaws in the Li and Ito method and the unknown time coefficient method are evident from the most simple of methods. In both of these methods, oscillations rapidly appear in the approximate solution near the interface, and these oscillations rapidly spread through the domain (see figure 10). Even though the FTCS method is unconditionally unstable, errors begin to accumulate at a much greater rate in these two methods than in the Russell and Wang method, which appears to be accurate down to round off error (see figure 11).

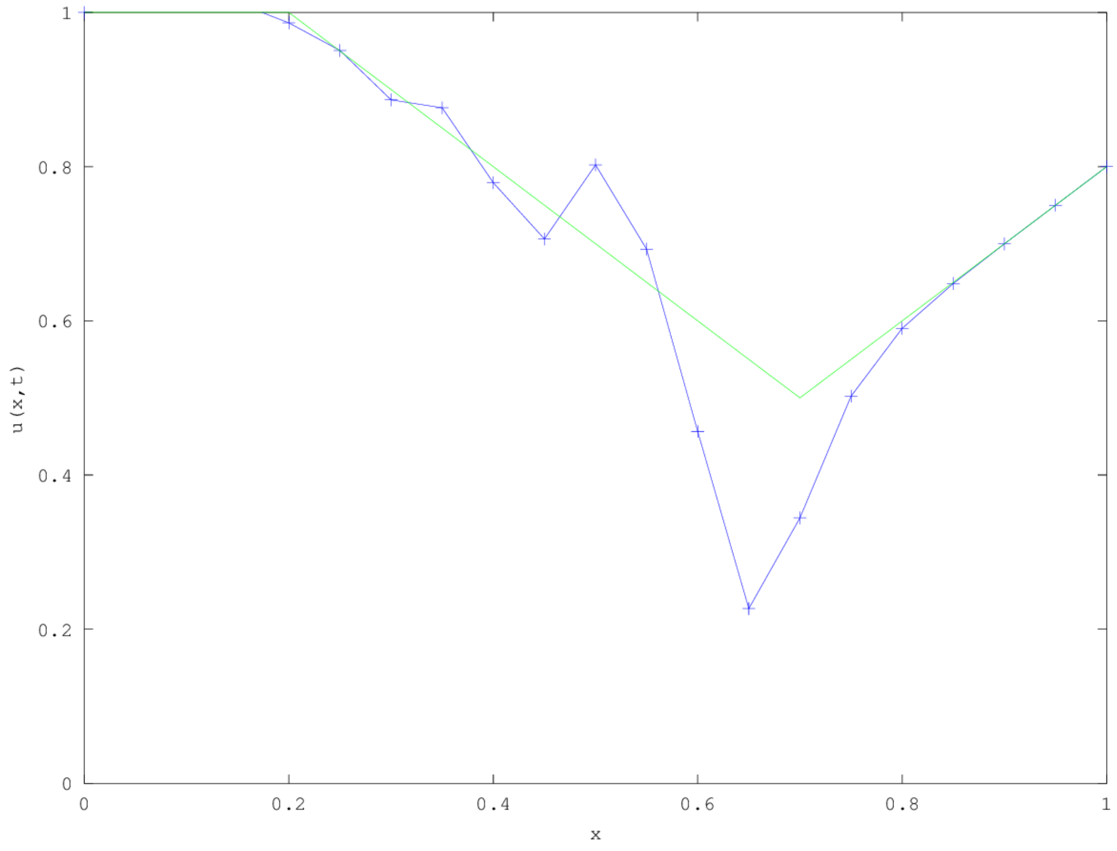
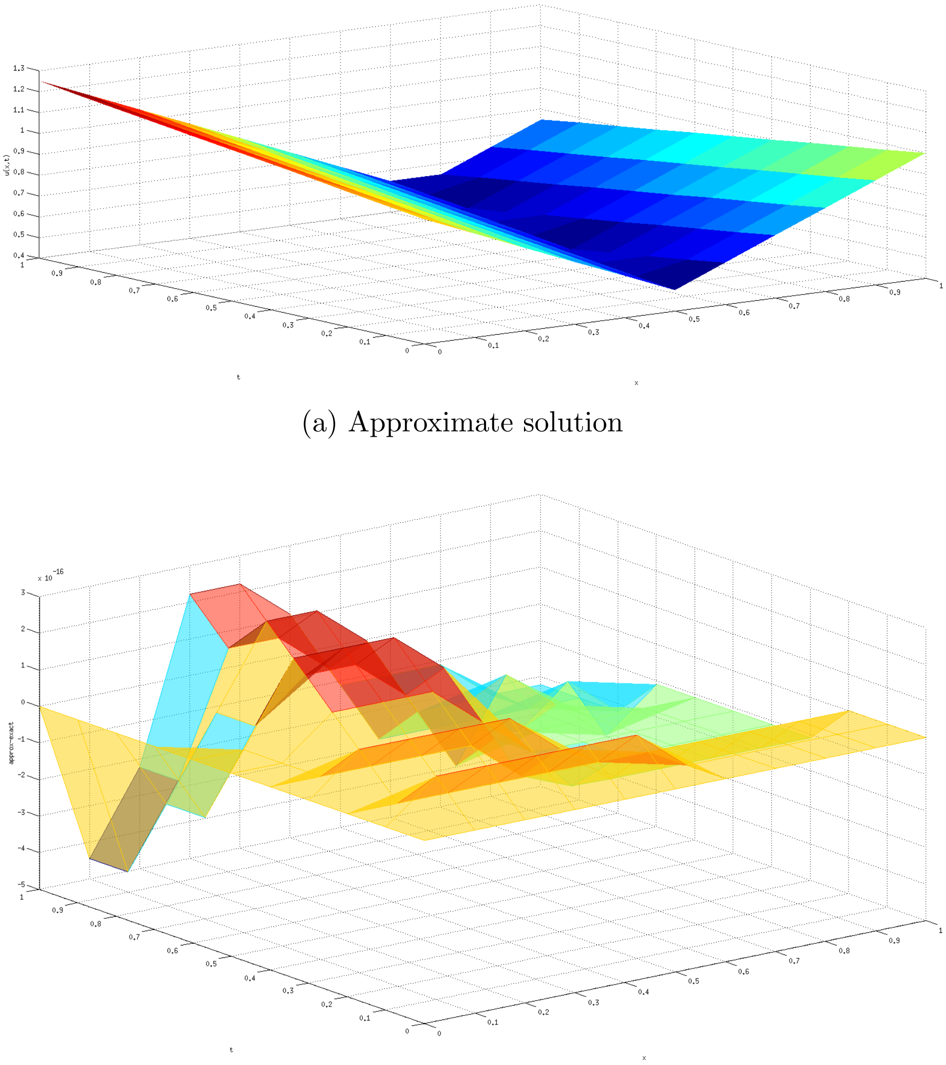


Figure 10: Approximate solution to the advection equation using the undetermined time coefficient method and FTCS finite difference scheme with $\Delta t = 0.01$, $\Delta x = 0.05$, $a = 1$, $t = 0.20$. The solid line represents the exact solution and the line with the × symbols represents the approximate solution.



(a) Approximate solution

(b) Errors

Figure 11: The results and errors computed using the Russell and Wang method applied to the FTCS finite difference scheme with wave-speed $a = 0.25$, $\Delta x = 0.1$, $\Delta t = 0.1$.

12.1.2 Backwards time differencing

The backwards time method shares the same error properties as the FTCS method ($\mathcal{O}(\Delta x^2)$ in space and $\mathcal{O}(\Delta t)$ in time) but it is unconditionally stable.

The backwards time method uses centered differences in space and backwards differences in time, making it an implicit method.

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} = -a\frac{u_{i+1}^{k+1} - u_{i-1}^{k+1}}{2\Delta x} \tag{68}$$

Despite its stability, the Li and Ito and unknown time coefficient methods perform equally poorly using this discretisation scheme, while the Russell and Wang method continues to perform well. Large errors begin to appear near the interface for the Li and Ito and unknown time coefficient methods soon after beginning the simulation. However unlike the FTCS method, these oscillations are relatively well damped through the rest of the domain most likely due to the implicit nature of this discretisation scheme.

12.1.3 Lax-Wendroff method

The Lax-Wendroff method is an $\mathcal{O}(\Delta x^2)$ in space and $\mathcal{O}(\Delta t^2)$ method in time method.

Its derivation involves taking a Taylor series approximation of $u(x_i, t_k + \Delta t)$ about t which gives

$$u(x_i, t_k + \Delta t) = u(x_i, t_k) + \Delta t\frac{\partial u}{\partial t}(x_i, t_k) + \frac{\Delta t^2}{2}\frac{\partial^2 u}{\partial t^2}(x_i, t_k) + \mathcal{O}(\Delta t^3) \tag{69}$$

Differentiating the advection equation with respect to t gives

$$\begin{aligned}
\frac{\partial}{\partial t}\left(\frac{\partial u}{\partial t}\right) &= \frac{\partial}{\partial t}\left(-a\frac{\partial u}{\partial x}\right) && (70)\\
\frac{\partial^2 u}{\partial t^2} &= -a\left(\frac{\partial^2 u}{\partial t \partial x}\right) && (71)\\
&= -a\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial t}\right) && (72)\\
&= -a\frac{\partial}{\partial x}\left(-a\frac{\partial u}{\partial x}\right) && (73)\\
&= a^2\frac{\partial^2 u}{\partial x^2} && (74)
\end{aligned}$$

Substituting $\partial u/\partial t$ and $\partial^2 u/\partial t^2$ into equation 69 gives

$$u(x_i, t_k + \Delta t) = u(x_i, t_k) - a\Delta t\frac{\partial u}{\partial x} + \frac{a^2\Delta t^2}{2}\frac{\partial^2 u}{\partial x^2} + \mathcal{O}(\Delta t^3) \tag{75}$$

Taking centered differences in time and truncating gives the Lax-Wendroff scheme

$$u_i^{k+1} = u_i^k - \frac{a\Delta t}{2\Delta x}\left(u_{i+1}^k - u_{i-1}^k\right) + \frac{a^2\Delta t^2}{2\Delta x^2}\left(u_{i+1}^k - 2u_i^k + u_{i+1}^k\right) \tag{76}$$

As with the other discretisation schemes, this is accurate away from the interface, but requires correction near the interface using one of the three correction schemes.

The Li and Ito and unknown time coefficient methods fare better with the Lax-Wendroff method, but they still cannot match the accuracy of the Russell and Wang method. Errors in the Li and Ito and unknown time coefficient methods are more localized to point surrounding that interface than in previous methods, but there is still some clear 'rippling' on the trailing edge of the discontinuity (see figure 12). In addition, the errors at the discontinuity are much larger than the expected order in both methods.

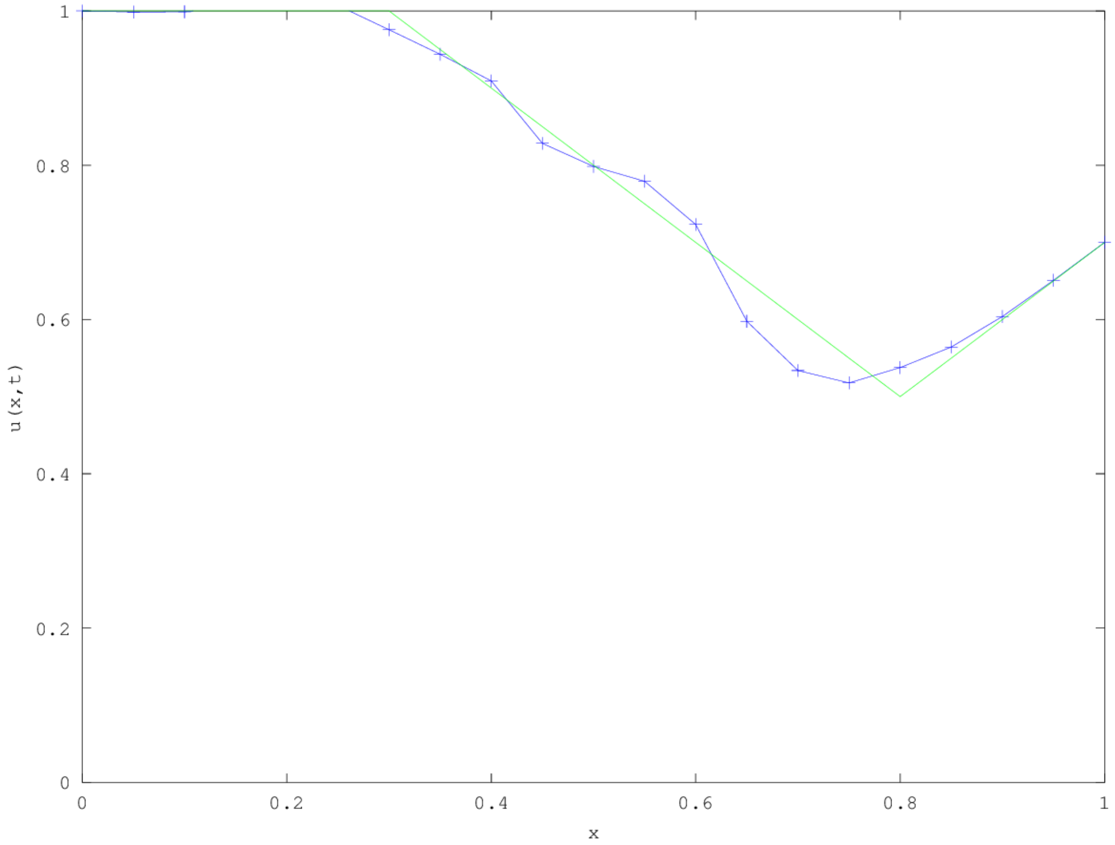


Figure 12: Approximate solution to the advection equation using the undetermined time coefficient method and a Lax-Wendroff discretisation. Parameters were $\Delta t = 0.01$, $\Delta x = 0.05$, $a = 1$, $t = 0.30$. The solid line represents the exact solution and the line with the × symbols represents the approximate solution.

12.2 The IIM for a Heaviside initial condition

Another test problem that was used to evaluate the accuracy of these three correction methods was identical to the previous problem but with the initial condition being a Heaviside function

$$u(x,0) = \begin{cases} 1 & x < \alpha(0) \\ 0 & x \geq \alpha(0) \end{cases} \tag{77}$$

which has exact solution

$$u(x,t) = \begin{cases} 1 & x < \alpha(t) \\ 0 & x \geq \alpha(t) \end{cases} \tag{78}$$

This problem produced a similar pattern of errors to the previous problem.

The Li and Ito and undetermined time coefficient methods both produced fairly poor results, with large oscillatory errors evident in both the FTCS and backwards time schemes (see figure 13). These two methods produced slightly better results using the Lax-Wendroff discretisation scheme, however there was still a large 'hump' that developed quickly on the trailing edge of the discontinuity.

As with the wedge initial condition, the Russell and Wang method proved to be accurate to the correct order with all discretisation schemes.

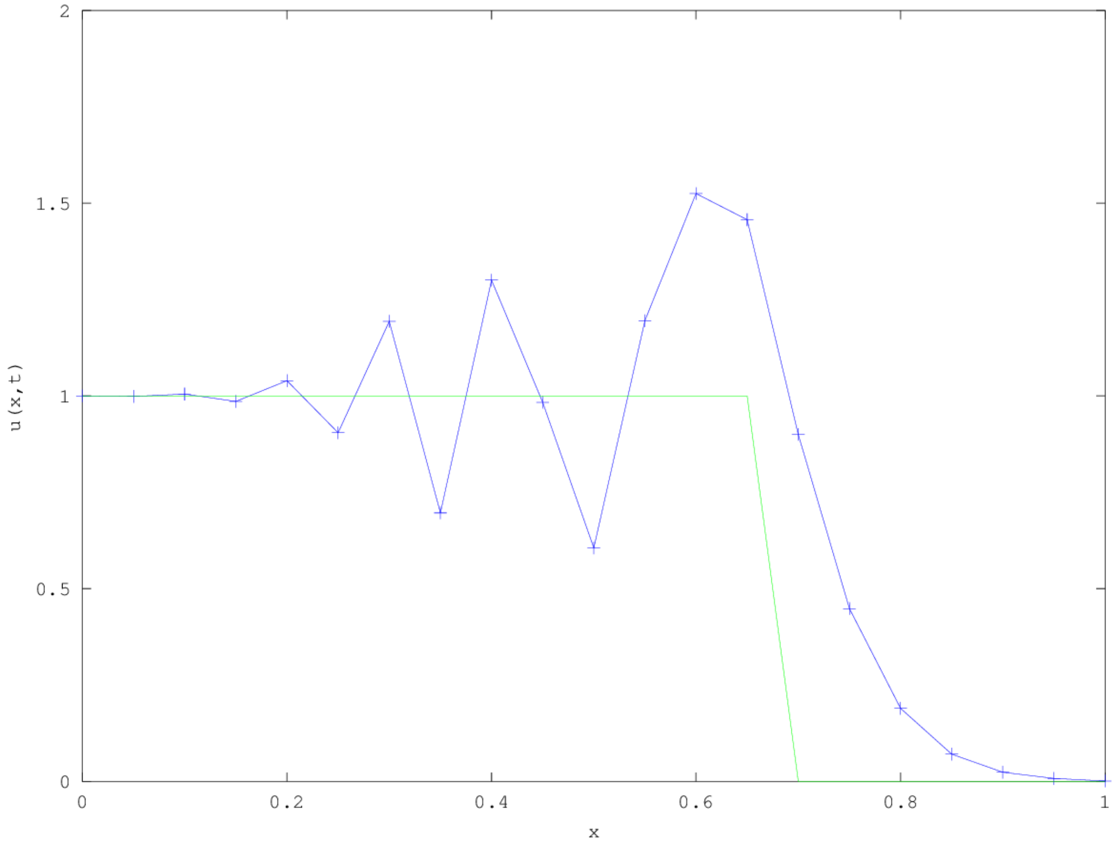


Figure 13: Approximate solution the advection equation using the Li and Ito method with a backwards difference discretisation. Parameters were $\Delta t = 0.01$, $\Delta x = 0.05$, $a = 1$, $t = 0.2$. The solid solution is the exact solution and the solution with the + symbols is the approximate solution.

13 Errors in the Li and Ito and undetermined time coefficient methods

The results obtained using the Li and Ito and undetermined time coefficient methods are not as accurate as was expected. Li and Ito present a problem in their book ([19] pp. 202-203) where they solve the heat equation with a moving interface using their discretisation scheme, and they are able to obtain an accurate result which is confirmed to be second order accurate in time and space using a grid refinement analysis. However my attempts to reproduce this result using an even simpler initial condition than was considered in this example did not result in success as shown in the previous section.

What this suggests is that although the Li and Ito method is fairly simple in its reasoning, the subtleties of implementation maybe be somewhat beyond what is discussed by Li and Ito in their book.

The undetermined time coefficient method was an attempt to produce accurate results after the failure of the Li and Ito for the simple problems considered in this project. Much time was spent attempting to correct the Li and Ito method before moving on to the undetermined coefficient method, and further time was spent attempting to correct this method. Although these attempts were ultimately unsuccessful, some patterns were observed in the behavior of errors of the undetermined coefficient method.

The first observation is related to the phenomenon of large 'spikes' occurring in the errors of the undetermined coefficient method when solving the advection equation with backwards differences and the wedge initial condition. Although accuracy of this method was ultimately poor, there appeared to be certain timesteps where the magnitude of the error would suddenly jump before returning to its normal rate of increase. By plotting the magnitude of the error against time, it was found that the errors tended to 'spike' when the value of $(\alpha - x_j + 1)$ became small. That is, when the interface became very close to the right hand irregular point.

The second observation is also related to the phenomenon of error 'spikes'. For some combinations of Δt and Δx it was observed that the timestep before a large spike, the condition number of the coefficient matrix of the system was observed to greatly increase, before returning to normal levels after the spike occurred. Interestingly, this ill-conditionedness of the matrix could be eliminated by reducing the size of Δt, but unfortunately the large error 'spikes' persisted despite the now consistently small condition number of the coefficient matrix.

14 Conclusions and future directions for the IIM

The IIM is a robust and capable method for solving differential equations involving interfaces. It is based off of the idea of correcting the finite difference scheme only at points near the interface by using Taylor series to ensure that accuracy is preserved. This allows it to be of similar computational difficulty as standard methods, but it enjoys far greater accuracy when discontinuities and singularities are present in a problem. This premise is relatively simple, but this simplicity makes it easy to modify and adapt the ideas presented in the original papers on the IIM to solve new and challenging problems or to improve the method in both accuracy and scope.

The IIM has been shown to be applicable to the three major classes of differential equations in one or many spatial dimensions and with moving interfaces. The IIM also has a number of related methods such as the EJIIM, the MIB and finite volume based methods that attempt to improve or adapt the IIM while retaining its ability to accurately resolve discontinuities. A variety of applications of the IIM in the physical and natural sciences have been presented, all of which use the IIM to solve novel problems that would otherwise be difficult to solve due to discontinuous phenomena.

Three different methods of correcting the IIM near the interface have been investigated. Although two of these methods proved to be inaccurate, the Li and Ito method has been shown to be accurate by others, and the undetermined coefficient method is entirely derived from the basic IIM which has also been shown to be accurate. This failure to implement a mathematically sound method exposes a potential downside to IIM based methods.

Although the IIM is accurate, its reliance on locating and correcting the solution at the points where the interface crosses the grid is both time-consuming and difficult. The analysis in one-dimension for monotonically behaving interfaces is not difficult, however in two or three dimensions with non-convex interfaces this analysis could be difficult or computationally prohibitive.

Some interesting behaviors in the errors of the Li and Ito and undetermined time coefficient methods were observed, and in the future it might be possible to ascertain the sources of these errors and implement these methods correctly.

The Russell and Wang method proved to be the most effective method for solving interface problems, even though all of these methods should produce results of equal accuracy.

The IIM has shown some promising developments in recent years which may warrant further research.

Berthelson [4] and Zhao et. al. [41] have shown that it is possible to construct methods for elliptic problems where the coefficient matrix is symmetric. This allows for fast linear solvers to be used with relative ease, and could greatly decrease the amount of computation required to solve elliptic interface problems.

Liang et. al. [24] and Jiang et. al. [10] have developed a spectral method for use with interface problems that retains accuracy (ie. does not produce Gibbs phenomena) despite discontinuities. This method appear to be promising as it will allow for the nice error properties of spectral methods when solving interface problems.

Despite the IIM being developed for hyperbolic problems relatively soon after its creation, there is a dearth of papers when compared with elliptic and parabolic problems. One possible explanation for this is given in section 3.3 but it may be prudent to investigate the IIM for hyperbolic problems further to see whether there are any issues in implementation that are specific to hyperbolic problems.

Acknowledgments

I would like to acknowledge Michael Page for his generous assistance and guidance throughout this year, and for allowing me to use his code to produce the results for the Russell and Wang method. I would also like to acknowledge my friends and family for their endless support and encouragement.

References

[1] Loyce Adams and Zhilin Li. The immersed interface/multigrid methods for interface problems. *SIAM Journal on Scientific Computing*, 24:463–479, 2002.

[2] Juan Ruiz Alvarez, Jinru Chen, and Zhilin Li. The IIM in polar coordinates and its application to electro capacitance tomography problems. *Numerical Algorithms*, 57:405–423, 2011.

[3] Attila J Bergou, Sheng Xu, and Z Jane Wang. Passive wing pitch reversal in insect flight. *Journal of Fluid Mechanics*, 591:321–337, 2007.

[4] Petter Andreas Berthelson. A decomposed immersed interface method for variable coefficient elliptic equations with non-smooth and discontinuous solutions. *Journal of Computational Physics*, 197:364–396, 2004.

[5] J W Cahn and J E Taylor. Surfae motion by surface diffusion. *Acta Metallurgica et materialia*, 42(4):1045–1063, 1994.

[6] Shaozhong Deng, Kazufumi Ito, and Zhilin Li. Three-dimensional elliptic solvers for interface problems and applications. *Journal of Computational Physics*, 184:215–243, 2003.

[7] Miguel A Dumett and James P Keener. An immersed interface method for solving anisotropic elliptic boundary value problems in three dimensions. *SIAM Journal on Scientific Computing*, 25:348–367, 2003.

[8] Aaron L Fogelson and James P Keener. Immersed interface methods for neumann and related problems in two and three dimensions. *SIAM Journal on Scientific Computing*, pages 1630–1654, 2000.

[9] J J Heys, T Gedeon, B C Knott, and Y Kim. Modeling arthropod filiform hair motion using the penalty immersed boundary method. *Journal of Biomechanics*, 41:977–984, 2007.

[10] Yongsong Jiang, An Liang, Xiaofeng Sun, and Xiaodong Jing. Multi-domain spectral immersed interface method for solving elliptic equation with a global description of discontinuous functions. *Chinese Journal of Aeronautics*, 25:297–310, 2012.

[11] R Kafafy, T Lin, Y Lin, and J Wang. Three-dimensional immersed finite element methods for electric field simulation in composite materials. *International Journal for Numerical Methods in Engineering*, 64:940–972, 2005.

[12] J S Langer. Instabilities and pattern formation in crystal growth. *Reviews of modern physics*, 52(1), 1980.

[13] Randall J Leveque. CLAWPACK - a software package for solving multi-dimensional conservation laws. In J Glimm, editor, *Proceedings of the 5th international conference on hyperbolic problems: theory, numerics, application*, pages 188–197, 1993.

[14] Randall J Leveque and Zhilin Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis*, 31:1019–1044, 1994.

[15] Zhilin Li. *The immersed interface method: A numerical approach for partial differential equations with interfaces.* PhD thesis, University of Washington, 1994.

[16] Zhilin Li. A fast iterative algorithm for elliptic interface problems. *SIAM Journal on Numerical Analysis*, 35:230–254, 1998.

[17] Zhilin Li. The immersed interface method using a finite element formulation. *Applied Numerical Mathematics*, 27:253–267, 1998.

[18] Zhilin Li and Kazufumi Ito. Maximum principle preserving schemes for interface problems with discontinuous coefficients. *SIAM Journal on Scientific Computing*, 23:339–361, 2001.

[19] Zhilin Li and Kazufumi Ito. *The Immersed Interface Method: Numerical Solutions of PDEs Involving Interfaces and Irregular Domains.* Society for Industrial and Applied Mathematics, 2006.

[20] Zhilin Li, Tao Lin, and Xiaohui Wu. New cartesian grid methods for interface problems using the finite element formulation. *Numerische Mathematik*, 96:61–98, 2003.

[21] Zhilin Li and A Mayo. Adi methods for heat equations with discontinuities along an arbitrary interface. *Proceedings of symposia in applied mathematics*, 48:311–315, 1993.

[22] Zhilin Li and Bharat Soni. Fast and accurate numerical approaches for stefan problems and crystal growth. *Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology*, 35:461–484, 1999.

[23] Zhilin Li, Hongkai Zhao, and Huajian Gao. A numerical study of electro-migration voiding by evolving level set functions on a fixed cartesian grid. *Journal of computational physics*, 152:281–304, 1999.

[24] An Liang, Xiaodong Jing, and Xiaofeng Sun. Constructing spectral schemes of the immersed interface method via a global description of discontinuous functions. *Journal of Computational Physics*, 227:8341–8366, 2008.

[25] Charles S. Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10:252–271, 1972.

[26] Charles S. Peskin. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25:220–252, 1977.

[27] Charles S. Peskin. The immersed boudary method. *Acta Numerica*, pages 478–517, 2002.

[28] David Russell and Z Jane Wang. A cartesian grid method for modelling multiple moving objects in 2d incompressible viscous flow. *Journal of Computational Physics*, 191:177–205, 2003.

[29] Vita Rutka. A staggered grid-based explicit jump immersed interface method for two-dimensional stokes flows. *International Journal for Numerical Methods in Fluids*, 57:1527–1543, 2008.

[30] Vita Rutka and Zhilin Li. An explicit jump immersed interface method for two-phase navier-stokes equations with interfaces. *Computing Methods for Applied Mechanical Engineering*, 197:2317–2328, 2008.

[31] James A. Viecelli. A method for including arbitrary external boundaries in the mac incompressible fluid computing technique. *Journal of Computational Physics*, 4:543–551, 1969.

[32] James A. Viecelli. A computing method for incompressible flows bounded by moving walls. *Journal of Computational Physics*, 8:119–143, 1971.

[33] J. E. Welch, F. H. Harlow, J. P. Shannon, and B. J. Daly. The MAC method: A computing technique for solving viscous, incompressible, transient fluid flow problems involving free surfaces. Technical Report LA-3425, Los Alamos Scientific Laboratory, 1966.

[34] Andreas Wiegmann. *The Explicit Jump Immersed Interface Method and Interface Problems for Differential Equations*. PhD thesis, University of Washington, 1998.

[35] Andreas Wiegmann and Kenneth P Bube. The immersed interface method for nonlinear differential equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis*, 35:177–200, 1998.

[36] Andreas Wiegmann and Kenneth P Bube. The explicit-jump immersed interface method: Finite difference methods for pdes with piecewise smooth solutions. *SIAM Journal on Numerical Analysis*, 37:827–862, 2000.

[37] Alexander Yakhot, Leopold Grinberg, and Nikolai Nikitin. Modeling rough stenoses by an immersed-boundary method. *Journal of Biomechanics*, 38:1115–1127, 2005.

[38] Trevor York. Status of electircal tomography in industrial applications. *Journal of electronic imaging*, 10(3):608–619, 2001.

[39] Chaoming Zhang. *Immersed interface methods for hyperbolic systems of partial differential equations with discontinuous coefficients*. PhD thesis, Univeristy of Washington, 1996.

[40] Chaoming Zhang and Randall J Leveque. The immersed interface method for acoustic wave equations with discontinuous coefficients. *Wave Motion*, 25:237–263, 1997.

[41] Jianping Zhao, Yanren Hou, and Yongfei Li. Immersed interface method for elliptic equations based on a piecewise second order polynomial. *Computers and Mathematics with applications*, 63:957–965, 2012.

[42] Shan Zhao and GW Wei. High-order fdtd methods via derivative matching for maxwell's equations with material interfaces. *Journal of Computational Physics*, 200:60–103, 2004.

[43] YC Zhou and GW Wei. On the fictitious-domain and interpolation formulations of the matched interface and boundary (MIB) method. *Journal of Computational Physics*, 219:228–246, 2006.

[44] YC Zhou, Shan Zhao, Michael Feig, and GW Wei. High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources. *Journal of Computational Physics*, 213:1–30, 2006.