

**A Hybrid Laplace Transform/Finite Difference Method for
Diffusion Problems**

Investigation of Inverse schemes.

Colin L Defreitas

Submitted to the University of Hertfordshire in partial fulfilment of the
requirement of the degree of Doctor of Philosophy.

Department of Physics, Astronomy and Mathematics University Of
Hertfordshire
United Kingdom
June 2021

Acknowledgments

This thesis has been a complex and challenging task. Spanning ten years, punctuated by periods of illness and recovery, it would not have been possible without the help and support from close friends and colleagues. As such I wish to thank my supervisor Dr Steve Kane whose encouragement, calm, good humour and insightful mathematical guidance kept me going to the end. I would also like to thank Dr Alan McAll, Dr Catarina Carvahlo and Carrie Ricketts for their helpful advice. I particularly want to thank my wife, Adeola King, who continuously encouraged me through many difficult times. I would also like to thank my friend Dr Nutan Rajguru for her helpful advice and pointed criticisms. A special thanks to Kathy Lee in the research office, whose advice and patience served me well throughout the process. And finally, to thank my very close friends Lester P Cassimy and Sydney Michael Vieira for their friendship and support whenever I needed it.

Abstract

The thesis carries out a series of investigations to improve the understanding and efficiency of the Laplace Transform Finite Difference Method (LTFDM).

In chapter two, I begin by investigating the noise handling properties of the Fourier series and the Stehfest and Talbot algorithms for inverting the Laplace transform. Here noise is added to various test functions, and the results are compared to the exact solutions. I find that the Talbot algorithm successfully reconstructs the function while both the Stehfest and the Fourier series methods fail to invert these functions accurately.

Chapter three extends the investigation by examining the performance of five of the main algorithms for inverting the Laplace transform in standard 16 digits precision and multi-precision. The results show that Talbot generally outperforms the other algorithms in standard precision while the Stehfest is the best in multi-precision.

The LTFDM is then used to solve the Fisher KPP (Kolmogorov, Petrovsky and Piskunov) equation. This equation has a travelling wave solution, and Fourier and Laplace transform numerical methods have difficulty reconstructing travelling waves. Using the knowledge gained in chapters two and three and understanding the nature of the perturbations in this equation, accurate representations of several solutions to this equation were produced.

The LTDFM is then successfully applied to a series of linear and non-linear diffusion equations. Comparisons are then made with the Forward Time Central Difference and the Crank Nicholson methods, with the LTDFM showing advantages over these schemes in both time and accuracy.

Contents

1	Introduction	6
1.0.1	Brief Background	8
1.0.2	Layout	9
1.0.3	The Diffusion Equation	12
1.1	Finite Difference Method	13
1.2	The Laplace Transform	17
1.3	Literature Review	21
1.3.1	The Laplace Transform Boundary Element Method (LTBEM)	21
1.3.2	LTFDM	23
1.3.3	Numerically Inverting the Laplace Transform	24
2	The Noise Handling Properties of the Talbot Algorithm for Numerically Inverting the Laplace Transform	27
2.0.1	Introduction	27
2.1	The Laplace Transform	28
2.2	The Inverse Laplace Transform Perturbation and Precision	29
2.3	The Algorithms	30
2.3.1	The Fourier Series Method	31
2.3.2	The Stehfest Algorithm	33

2.3.3	The Talbot Algorithm.	36
2.3.4	Tests	43
2.3.5	Results	46
2.4	Summary.	56
3	The Numerical Inversion of the Laplace Transform in a Multi-Precision .	57
3.1	Introduction	58
3.2	The Laplace transform	59
3.3	The Inverse Laplace Transform, Perturbation and Multi-precision	60
3.4	The Algorithms	61
3.4.1	The Fourier Series Method	61
3.4.2	Gaver's Functional	62
3.4.3	Logan's Version Of The Talbot Algorithm	66
3.5	Results	68
3.6	Summary	76
4	A Laplace Transform Finite Difference Scheme for the Fisher-KPP Equation.	78
4.1	Introduction	78
4.2	Fisher's equation	79
4.3	The Laplace Transform Finite Difference Method	82
4.3.1	Inverting the data	83
4.4	Method	84
4.5	Numerical examples and discussion	89
4.6	Results	94
4.7	Summary	105

5	The Laplace Transform Finite Difference Method for Solving Linear and Non-Linear Diffusion Equations.	107
5.1	Introduction	107
5.1.1	The Laplace transform	109
5.1.2	Laplace Transform Finite Difference Method	110
5.2	The Inversion Algorithms	113
5.2.1	Stehfest	113
5.3	Diffusion Equations.	115
5.3.1	Problem 1	117
5.3.2	Problem 2	117
5.3.3	Problem 3	120
5.4	Non-Linear Diffusion Type problems	125
5.4.1	Problem 4	125
5.4.2	Problem 5	129
5.4.3	Problem 6	133
5.4.4	Problem 7	136
5.5	Summary	141
6	Conclusions and Future Work	143
6.0.1	Future Work	146
7	References	148

Chapter 1

Introduction

The diffusion process is a widely occurring natural phenomenon, and the mathematical expression of this process has been enormously successful in modelling diffusive behaviour. Thus, the diffusion equation is one of the most important partial differential equations in all mathematics and the applied sciences. As is the case with most partial differential equations, analytic solutions exist for only a small class of these problems [63], [12], [44], [85], [80] and numerical methods are continuously being developed to solve diffusion type problems [6], [71], [86], [115].

The most commonly used methods for numerically solving the diffusion equation usually employ a time stepping process that can impose stringent stability restrictions on the time step size. As an alternative, the Laplace transform can be used so that the solution at a particular time is not dependent on the solution at any other time, apart from the initial conditions. This gives the potential to attain the required solution in just one time step.

This research aims to improve the Stehfest and Talbot algorithms' understanding when using the Laplace transform to find numerical solutions to diffusion

type problems, emphasising the Laplace Transform Finite Difference method (LTFDM). This was done by testing the algorithm's response to noisy data, applying the algorithms within the context of the LTFDM and testing their performance with extended precision.

However, the drawback of using this method is that the Laplace transform's numerical inversion is known to be an ill-posed problem meaning that changes in the solution's behaviour are not continuous with the initial conditions so that small errors in the initial data can result in much larger errors in the solution [23],[26],[45], [64], [45],[61].

So the thesis also examines how the ill-posed nature of the numerical inversion of the Laplace transform might be mitigated to more successfully apply the LTFDM to solve diffusion type problems. To achieve these aims, I:

1. Tested the performance of three widely used algorithms used to invert the Laplace transform numerically; The Fourier Series method [28], The Stehfest algorithm [99] and the method developed by Talbot [104] for their noise handling properties, to discern which of these algorithms are best suited for handling errors in the input data. The study was new work as no systematic analysis of these algorithms' ability to cope with noise had been done before.
2. Extended the LTFDM to solve the reaction-diffusion Fisher-KPP equation. This equation typically results in a travelling wave profile for which Laplace transform methods are not usually employed. Accurate solutions to this problem were attained with some adjustments to the LTFDM and using our previous knowledge of the inversion schemes. Laplace transform methods had not been previously employed to solve this equation.
3. Investigated the performance of various forms of these algorithms in a multi-precision environment on different test functions to see what effect this had on reducing the ill-posed nature of the numerical inversion. This investigation al-

lowed us to determine which algorithm works best in standard 16-digit double precision and performed best in extended precision. This investigation extended and deepened a previous study [1] on the effects of multi-precision on numerically inverting the Laplace transform to include previously untested versions of the algorithms. Multi-precision computing allows the user to stipulate the number of significant digits needed for a particular calculation.

4. Employed the LTFDM to solve a wide variety of linear and non-linear diffusion type equations. The method was successfully tested on equations with various initial and boundary conditions and varying degrees of non-linearity. This demonstrated the successful application of the LTFDM to a wide variety of previously untested linear and non-linear parabolic partial differential equations with Dirichlet conditions.

1.0.1 Brief Background

Many advanced numerical methods are available for solving the diffusion equation. For the most part, they require the use of the Finite Difference Method (FDM), Finite Element Method (FEM), Boundary Element Method (BEM) and Finite Volume Method (FVM). These methods involve the discretisation of both the time and spatial variables. This combined discretisation introduces stringent stability criteria that limit the time step size used in these methods. While methods such as the Crank-Nicholson scheme introduce conditions that can improve the solution's stability for more extensive time steps, stability criteria involving incremental time stepping still need to be applied. The finite-difference solution at each time step can involve hundreds and sometimes thousands of matrix inversions to arrive at a solution. For non-linear cases, the problem is further compounded as another iterative step is usually required.

The Laplace transform can transform a function (or numerical data) from the time domain into the Laplace s domain. This transform allows for removing time marching procedures in the finite-difference schemes used to solve time-dependent parabolic partial differential equations. This gives us the potential to attain a solution at **any** desired time.

However, using the Laplace transform can generate data in the Laplace domain, which is not easily invertible to the real domain by analytical means, [61], [37]. Thus numerical inversion techniques have to be used to convert the Laplace domain to the time domain. However, the Laplace transform's numerical inversion is known to be ill-posed, so the output of the inversion depends discontinuously upon the initial conditions. This means that numerically inverting the Laplace transform is a perturbed problem, so errors introduced into the input data can cause oscillations in the output data making numerical solutions potentially unstable. While this perturbation cannot be removed entirely, it can be curtailed through the choice of the inversion method and by working in unlimited precision.

1.0.2 Layout

The thesis is set out as follows;

Chapter one

In chapter one, I contextualise the thesis by introducing some preliminary material on the heat equation, finite-difference schemes, the Laplace transform and a literature review.

Chapter two. This chapter investigated the noise handling properties of three of the most widely used algorithms for numerically inverting the Laplace transform. Here I examine the algorithms' genesis, and their error handling properties

are evaluated through a series of standard test functions in which noise is added to the inverse transform. Comparisons are then made with the exact data.

Chapter three

In this section, I apply a modified version of the LTFDM to solve the Fisher-KPP equation [48]. The travelling wave solutions usually associated with the Fisher-KPP equation are generally not deemed suitable for treatment using Fourier or Laplace transform numerical methods [8]. However, I was able to obtain accurate results when some degree of time discretisation and a reinitialising of the initial conditions were inbuilt into the process.

Chapter four

In this chapter, I examine the performance of five algorithms for numerically inverting the Laplace transform in standard, 16-digit and multi-precision environments to determine the effect of this might have on reducing the perturbations of the inverse transform. The algorithms are taken from three of the four main classes of numerical methods used to invert the Laplace transform [1].

Because the numerical inversion of the Laplace transform is a perturbed problem [23], [45], and [61], rounding errors that are generated in numerical approximations can adversely affect the accurate reconstruction of the inverse transform. This chapter demonstrates that working in a multi-precision environment can substantially reduce these errors and the resulting perturbations that exist in transforming the data from the s -space into the time domain, thus overcoming the main drawback of numerically inverting the Laplace transform.

Chapter five

In this chapter, the Laplace Transform Finite-Difference Method LTFDM is used to solve several linear and non-linear diffusion type problems with Dirichlet (or first-type) boundary conditions. This chapter uses the method to solve one dimensional linear and non-linear diffusion problems with various initial

and boundary conditions. For each of the equations considered, the time domain solution is provided via the numerical inversion Laplace transform using the algorithms proposed by Stehfest and Talbot. [99], [104]. The accuracy of the algorithms is then compared.

1.0.3 The Diffusion Equation

Fourier, one of the first people to study this phenomenon, showed through experimentation that an empirical relationship exists between the conduction rate in a material and the temperature gradient in the direction of energy flow. He concluded that “the heat flux resulting from thermal conduction is proportional to the magnitude of the temperature gradient and opposite to its sign”, [16]. The differential form expressing this relationship can be stated as,

$$q = -\alpha \Delta T \quad (1.1)$$

where with SI units

q is the local heat flux density, Wm^{-2}

α is the thermal conductivity, $\text{W}^{-1}\text{K}^{-1}\text{m}^{-1}$

and ΔT is the temperature gradient, Km^{-1}

For unidirectional conduction in the x direction then the equation can be expressed as

$$\mathbf{q}_x = -\alpha \frac{\partial T}{\partial x} \quad (1.2)$$

Consider a long thin bar of constant cross section which conducts heat uniformly throughout its length. Let the bar be perfectly insulated so that heat flows only laterally along the bar and the temperature distribution $T(x, t)$ depends only on its distance x along the bar at a time t . In this situation Fourier’s law gives the rate of heat flow $\mathbf{q}$ as

$$\mathbf{q} = -\alpha A \frac{\partial T}{\partial x} \quad (1.3)$$

where A is the cross-sectional area of the bar. Next consider an infinitesimally small section of the bar of length dx which has a rate of heat flow across its ends and q_2 with temperature distributions T at x and $T + \frac{\partial T}{\partial x} dx$ at $x + dx$

respectively. So the net heat flow across that section in an element of time dt is

$$= -\alpha A \left[\frac{\partial T(x+dx, t)}{\partial x} - \frac{\partial T(x, t)}{\partial x} \right] dt \quad (1.4)$$

$$= -\alpha A \left[\frac{\partial}{\partial x} \left(T + \frac{\partial T}{\partial x} dx \right) - \frac{\partial T(x, t)}{\partial x} \right] dt \quad (1.5)$$

$$= \alpha A \frac{\partial^2 T}{\partial x^2} dx dt \quad (1.6)$$

The conductor's specific heat under consideration is the total amount of heat required per unit mass per unit temperature. Hence the heat gained by the element in time dt is proportional to the mass mdx and the temperature increase in time dt .

$$\text{the heat gained in time } dt = c_p \rho A dx \frac{\partial T}{\partial t} dt \quad (1.7)$$

Where c_p and ρ is the specific heat capacity and density of the medium respectively. Equating (1.6) and (1.7) we get the heat equation

$$\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial x^2} \quad (1.8)$$

where $\kappa = \frac{\alpha}{\rho c_p}$ is the thermal diffusivity. In the literature the notation for temperature distribution $u(x, t)$ is usually used instead of $T(x, t)$. I follow this practice in this thesis. So in this form (1.8) becomes

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2} \quad (1.9)$$

1.1 Finite Difference Method

With the Laplace transform, I will incorporate finite-difference schemes to solve the diffusion equation. Most partial differential equations cannot be solved in

terms of explicit analytical formulas [12] [44] [85] [80] and so numerical methods have to be used for finding solutions to these equations. Besides the Finite Difference Method the most widely used numerical methods are the Finite Element Method (FEM), [9] the Finite Volume Method (FVM) [46], [108], and Boundary Element Method (BEM), [22], [22]. These methods all involve using domains modelled by mesh systems to discretise the region of integration. Many mesh-free numerical methods are used to solve certain types of partial differential equations,[53], [21]. This section looks at how the partial derivatives particularly, the spatial partial derivatives, are approximated. These finite-difference approximations are derived using Taylor series expansions. The basic idea can be seen in how the differential $\frac{dy}{dx}$ can be written in terms of an approximate finite-difference. If $y = f(x)$ then,

$$\frac{dy}{dx} = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \quad (1.10)$$

So if h is sufficiently small then

$$\frac{dy}{dx} \approx \frac{f(x+h) - f(x)}{h} \quad (1.11)$$

This is called the forward difference approximation and so the backward difference approximation is given by

$$\frac{dy}{dx} \approx \frac{f(x) - f(x-h)}{h} \quad (1.12)$$

A Taylor series expansion can provide expressions for various orders of partial derivatives accompanied by error bounds for these approximations. The Taylor series for the function $u(x, y)$ expanded about the point (x, y) in the x variable

is

$$u(x+h, y) = u(x, y) + h \frac{\partial u(x, y)}{\partial x} + \frac{h^2}{2!} \frac{\partial^2 u(x, y)}{\partial x^2} + \frac{h^3}{3!} \frac{\partial^3 u(x, y)}{\partial x^3} + \dots \quad (1.13)$$

where the spatial step h is sufficiently small to ensure that the series converges.

Similarly

$$u(x-h, y) = u(x, y) - h \frac{\partial u(x, y)}{\partial x} + \frac{h^2}{2!} \frac{\partial^2 u(x, y)}{\partial x^2} - \frac{h^3}{3!} \frac{\partial^3 u(x, y)}{\partial x^3} + \dots \quad (1.14)$$

These two expansions allows us to get the expressions for the partial derivatives, from (1.13) which can be truncated and so approximated by

$$\frac{\partial u(x, y)}{\partial x} = \frac{u(x+h, y) - u(x, y)}{h} + O(h) \quad (1.15)$$

and from (1.14) by

$$\frac{\partial u(x, y)}{\partial x} = \frac{u(x, y) - u(x-h, y)}{h} + O(h) \quad (1.16)$$

where $O(h)$ is the leading term of the truncation error given by

$$\frac{h}{2!} \frac{\partial^2 u(x, y)}{\partial x^2}$$

Also 1.13 and 1.14 can be rearranged to get the central difference scheme

$$\frac{\partial u(x, y)}{\partial x} = \frac{u(x+h, y) - u(x-h, y)}{2h} - \frac{h^2}{3!} \frac{\partial^3 u(x, y)}{\partial x^3} + \dots \quad (1.17)$$

written as

$$\frac{\partial u(x, y)}{\partial x} = \frac{u(x+h, y) - u(x-h, y)}{2h} + O(h^2) \quad (1.18)$$

If (1.13) and (1.14) are added and rearranged, the central difference for the second order derivative is obtained.

$$\frac{\partial^2 u}{\partial x^2} = \frac{u(x+h, y) - 2u(x, y) + u(x-h, y)}{h^2} + O(h^2) \quad (1.19)$$

These difference approximations can now be expressed using mesh (or grid notation) where $x = ih$ and $y = jk$ where h and k are the respective step sizes, to get

$$\frac{\partial^2 u_{i,j}}{\partial x^2} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + O(h^2) \quad (1.20)$$

It is then possible to develop several finite-difference approximations as forward, backward and central differences for higher order derivatives. Moreover it is possible to find expressions for them with higher order accuracy for example

$$\frac{\partial^2 u_{i,j}}{\partial x^2} = \frac{-u_{i+2,j} + 16u_{i+1,j} - 30u_{i,j} + 16u_{i-1,j} - u_{i-2,j}}{12h^2} + O(h^4) \quad (1.21)$$

[105], [82]. However, as there are more nodes to calculate, this can lead to unnecessarily long computation times, and the values of some of the nodal points, particularly on the boundary, may not be available to us. Thus most finite difference schemes which model the diffusion make extensive use of (1.20) for approximating the spatial derivative $\frac{\partial^2 u}{\partial x^2}$.

For general finite difference methods, both time and spatial dimensions exist. As such, convergence and stability issues arise as important factors for obtaining the correct solution of the governing partial differential equation. Convergence involves controlling the discretisation error (the difference between the exact solution and the finite difference approximation). The magnitude of this error at each mesh point depends on the size and relationship between the spatial mesh length δx and δt . In an explicit scheme such as the Forward Time Central Space (FTCS) we must have that $r = \frac{\delta x}{\delta t^2} \leq \frac{1}{2}$. For more stable implicit schemes

such as the Crank Nicholson method, which is valid for all finite values of r a large value will lead to an inaccurate solution, [96].

Then there are stability conditions to consider. This concerns the unbounded growth of truncation error which accompanies the time row advances from the initial line denoting the initial conditions to the final line denoting the desired time.

The critical point here is that because the Laplace transform can get rid of the time derivative, we can have more confidence in the convergence and stability of our method as we have no restrictions on the time step.

1.2 The Laplace Transform

The project involves the use of the Laplace transform so that the time development is obtained using a hybrid Laplace Transform Finite Difference Method LTFDM. The Laplace transform is an integral transform defined as follows: Let $f(t)$ be defined for $t \geq 0$, then the Laplace transform of $f(t)$ is given by,

$$\mathcal{L}\{f(t)\} = \int_0^{\infty} f(t)e^{-st} dt \quad (1.22)$$

Thus $\mathcal{L}\{f(t)\}$ is a function of s denoted as $F(s)$. The Laplace transform can be shown to exist for any function which can be integrated over any finite interval $0 < t < l$ for $l > 0$, and for which $f(t)$ is of exponential order, i.e.

$$|f(t)| < Me^{at} \quad (1.23)$$

as $t \rightarrow \infty$, where $M > 0$ is a finite real number and a is a small real positive number.

The recovery of the function $f(t)$ is via the inverse Laplace transform which is

most commonly defined via the Bromwich contour integral. [98]

$$L^{-1}\{F(s)\} = f(t) = \frac{1}{2\pi i} \int_{\alpha-i\infty}^{\alpha+i\infty} f(s) e^{st} ds \quad (1.24)$$

However, using the Laplace transform can generate data in the Laplace domain, which is not easily invertible to the real domain by analytical means. Thus numerical inversion techniques have to be used to convert the data from the s domain to the time domain. The main problem with the numerical inversion of the Laplace transform is that it is known to be an ill posed or perturbed problem (thus, small changes in the input function can lead to large oscillations in the solution) [23],[45],[60]. to illustrate this, consider Bellman and Roth's analysis [10] of the Laplace transform and its inversion.

They begin by examining the choice of s in (1.22). If s is allowed to be any arbitrary complex parameter, then by (1.22), the function $\bar{f}(s)$ will also be complex. This situation leads to solving two integral equations, which involve singularities and oscillations for $s \leq 1$. To demonstrate this consider again the Laplace transform,

$$L\{f(t)\} \equiv \bar{f}(s) = \int_0^\infty f(t) e^{-st} dt \quad (1.25)$$

and s to be any complex number. As s is complex then $F(s)$ must be a function of a complex number. From a numerical standpoint the limits of integration of the Laplace Transform can create difficulties in evaluating the integral. Thus they apply the transformation $r = e^{-t}$ and the integral becomes,

$$- \int_0^1 f(-\ln r) \frac{r^s}{r} dt = \int_0^1 g(r) r^{s-1} r dt \quad (1.26)$$

where $g(r) = f(-\ln t)$

As $s \in \mathbb{C}$ then

$$r^{s-1} = r^{u-1-iv} = r^{u-1} e^{iv \ln r}. \quad (1.27)$$

Now $F(s) = a(uv) + ib(uv)$ thus

$$a(uv) = \int_0^1 r^{u-1} \cos(v \ln r) g(r) dr \quad (1.28)$$

$$b(uv) = \int_0^1 r^{u-1} \sin(v \ln r) g(r) dr \quad (1.29)$$

There are two integral equations to solve. More importantly, if $u < 1$, there will be a singularity at $r = 0$, making numerical computation difficult if not impossible. Also, if $v \neq 0$ aside from the rapid oscillations at the origin, the integral is unbounded [10]. In (1.22) and (1.24), Bellman and Roth choose s so that the singularities of $F(s)$ lie to the left of the $\text{Re}(s) = \alpha$ in the complex s -plane. Bellman and Roth conclude, “To avoid these problems s is chosen to lie on the positive real axis greater than unity”. The problem with this approach is that any numerical quadrature of the integral in (1.26) leads to an ill-conditioned matrix equation and hence instabilities in numerical inversion. As I show in this thesis adopting procedures that treat s in (1.22) and (1.24) as complex can reduce the effects of perturbation on the inverse transform.

Also, a cursory look at (1.22) shows the effect the e^{-st} term can have on smoothing out the values of $f(t)$ for large t making the recovery of this ‘lost information’ from the inverse transform difficult. A similar examination of (1.24) shows that the e^{st} term in the integral can amplify small changes in the input data, clearly fulfilling the requirement of being ill-posed. However, because of Bellman and Roth’s points, using complex arithmetic cannot completely overcome the problem’s ill-posed nature.

There have been other attempts to explain the ill-posed nature of the numerical

inversion of the Laplace Transform. Craig and Thompson [26] explain this as being inherent to forward and backward problems, noting that the instability of the Laplace inversion is the price that has to be paid for favourable smoothening properties of the forward transform in (1.24). They also point out that “The trade-off between frequency resolution and stability in the numerical solution is a recurring feature of classical approaches to ill-posed problems”.

Epstein and Schotland [45] use harmonic analysis relevant to forward and backwards transforms and demonstrate the inherently perturbed nature of the problems in handling noisy data. McWhirter and Pike [70] examine the ill-posed nature of numerically inverting the Laplace transform from the standpoint of Fredholm integral equations of the first kind. This class of equations includes the Laplace transform. Again they show how successful inversion depends on the noise level in the calculation and “on the eigenvalue structure of the kernel”. Despite these drawbacks, the Laplace transform’s power is apparent from its numerous applications spanning all areas of applied sciences. Thus, the above authors’ considerations were to deepen the understanding of its implementation to further its use in solving differential equations. Its ability to construct frequency domain representation converting (the description of the system from the time domain to the frequency domain) the differential equations to algebraic equations, which are much easier to solve. As I show in this thesis, when numerical Laplace transform methods are employed to solve these equations, accurate representation of the solution is possible, posing a viable alternative to spatial discretisation methods used for solving time-dependent parabolic partial differential equations.

1.3 Literature Review

In this section, I review important contributions in the literature which helped inform the thesis. This covers using Laplace transform methods to solve diffusion type equations and the principal methods used to invert the data numerically.

1.3.1 The Laplace Transform Boundary Element Method (LTBEM)

Moridis and Reddel [74] introduced the Laplace Transform Boundary Element, LTBEM, to simulate two and three-dimensional heat conduction and groundwater flow. Significantly, the authors produced results more accurately than the conventional BEM treatment.

Moridis [73] later developed an alternative formulation of the LTBEM in which two time-marching schemes were compared with the LTBEM. [29], [15]. The inversion from the Laplace space was done using the algorithms proposed by Stehfest [99], and DeHoog [37]. Moridis concluded that the DeHoog algorithm was less computationally efficient than the Stehfest inversion but allowed solutions at a range of times. He concluded that for this application, “The Stehfest LTBEM seems to have a clear advantage, except in cases involving very steep functions of time”.

Stradhar et al. [101] used the LTBEM to investigate three-dimensional problems involving transient heat conduction in homogeneous and non-homogeneous materials. The numerical simulation is done using a Galerkin approximation [14], and the time dependence is restored via the Stehfest algorithm. The authors report that the results were in excellent agreement with the analytical solutions

on the test problems tried using this method.

Zhu et al. published three papers exploring the combined Laplace transform and dual reciprocity method (LTDRM) for solving linear and non-linear diffusion equations [117], [119], [118]. Their work showed the combined scheme's ability to produce efficient, accurate numerical solutions for linear and non-linear diffusion type problems at any desired observation time.

Singh et al. [95] employed a mesh-free numerical method based on the particular solution for the modified Helmholtz operator and then used the Laplace transform to eliminate the time parameter. Describing the approach as a 'time free' 'mesh-free' method, the authors report better performance than the LTDRM for the problems examined in the paper. The Stehfest inversion scheme was used for this method.

For some years, members of the mathematics department at The University of Hertfordshire have investigated some of the various solution methods for solving the diffusion equation [31], [34], [56]. In continuing this research, the Laplace transform was used as a viable strategy for solving these equations by removing the time dimension and its associated time stepping procedures.

Crann's 2005 PhD thesis [27] investigated the use of the Laplace Transform combined with the Boundary Element (LTBEM) method for solving diffusion type problems. Radford's 2008 thesis, [87] looked at aspects of the Laplace Transform Isotherm Migration Method, and Fitzharris's thesis, [49] used Laplace transform methods to solve the diffusion type Black Scholes equation [13] .

In 2007 Kane et al. [58] investigated using a Hybrid Laplace Transform/Finite Difference Boundary Element Method for Diffusion Problems. All the above research at Hertfordshire used a time-domain decomposition suitable for implementation in a parallel computing environment. The main focus was on the issues associated with parallel computing when using Laplace transform meth-

ods. In all cases, the Stehfest inversion scheme was used to invert the data.

1.3.2 LTDFM

Opaunga et al. used the LTDFM on one-dimensional boundary value diffusion problems. Their results were compared to finite-difference methods, but their method involved using closed-form solutions and did not use numerical inversion algorithms [81]. Tagliani et al. used the technique of mixing the Laplace Transform and the finite-difference method to solve the Black-Scholes PDE [103], [13] in which the Post-Widder formula was used to invert the data [23],[83].

This method for solving the Black-Scholes equation was also used by Ann et al. [4] who used an algorithm developed by the authors but based on the Fourier series inversion method [3].

Chen et al. applied the LTDFM to a two-dimensional non-linear heat conduction problem using the Honig and Hirdes version of the Fourier series method for inverting the Laplace Transform. [55], [20].

Jingtang Ma et al. used the LTDFM in the context of “Pricing American options under complex models” [67] and employed the Stehfest inversion method. Zahra et al. used an LTDFM for obtaining “solutions fractional-order electrical circuits”[116]. As in [81] closed form solutions of the Laplace transform were used with no numerical inversion schemes employed.

Habte et al. [54] successfully applied the LTDFM for the specific application of “simulating the pressure-transient behaviour of oil/water flow associated with water injection/falloff tests”. The method is “coupled with the well known Buckley-Leverett frontal-advance formula [30] to solve the radial diffusivity equation”.

Mahajerin [68] combined an extension of the Fundamental Collocation Method

“to handle two-dimensional transient heat conduction problems in solids”. The method is applied in the Laplace space, and an inversion technique [91] is used to retrieve the solution in time. The authors noted that there were “inherent advantages over the domain-oriented techniques like the finite element and, finite difference methods, the Laplace transform-based FCM approach presented here may be regarded as a simpler method for solving a wide variety of time-dependent problems in heat conduction and related fields”.

1.3.3 Numerically Inverting the Laplace Transform

Davies and Martin [36] give a detailed account of their tests on 14 methods for numerically inverting the Laplace transform. They do so through a variety of test functions. “The methods are presented briefly and classified theoretically into methods that compute a sample, methods which expand $f(t)$ in exponential functions, methods based on Gaussian quadrature, methods based on a bilinear transformation, and methods based on Fourier series”. The conclusions are wide-ranging but note that the Stehfest and the Fourier series methods gave good results for most of the functions tested. The survey did not, however, evaluate the performance of the Talbot algorithm [104].

Other surveys include; Narayanan and Beskos, [78] who examined eight existing methods for numerically inverting the Laplace Transform. They confirm that using the Laplace transform for time-dependent diffusion equations offers a simple, straightforward and uniform method of solution by reducing the complexity of the problem through the reduction by one of the numbers of their independent variables. At the same time, “easily handling time-dependent boundary conditions”. They say the choice of method can depend on a compromise between methods with low accuracy but short computation times and those that

offer high accuracy but greater computation times.

Kuhlman [60], used a two dimensional BEM to compare five methods for inverting the Laplace Transform. He found that for this particular application, the “Fourier-series-based inversion algorithms work for common time behaviours, are the most robust with respect to free parameters, and allow for straightforward image function evaluation re-use across at least a log cycle. Of time.”

Cohen [23] provides a very useful summary of the derivation of the most widely used algorithms for numerically inverting the Laplace Transform, illuminating further the advantages and disadvantages of using an algorithm in a variety of applications.

Duffy [40], compares three methods; the method of Weeks [111], the Talbot method [104], and the Laguerre polynomial scheme [40] on a series of functions found in engineering and physics. Duffy concludes “that all the methods give good results, and the exact choice will depend on the problem”. He also noted that the Talbot algorithm is very fast.

Abate and Valko investigated the performance of the Gaver-Wynn-Rho (GWR) and the Talbot algorithm in a multi-precision environment [1]. They found that both had a greatly improved performance as the extra precision could curtail the inherent perturbation of the numerical inversion of the Laplace transform.

Wang and Zhan [109] ran tests on seven different inversion methods. Four of these were versions of the Fourier series method; the other three were Weeks, Talbot and Stehfest. They were used on solute transport problems. They concluded that Talbot, de Hoog and Simon worked very well for radial dispersion methods “regardless of the dispersion-dominated or advection-dominated problems”.

Krougly et al. [59] further examined the performance of the Stehfest algorithm in a multi-precision environment. After noting the improved performance of the

algorithm in this environment, the authors concluded: "We demonstrated that the level of precision chosen must match algorithms properly. In the Gaver-Stehfest algorithm, the balance is between the truncation error" (a consequence of truncating the series representation of the algorithm) " and roundoff error". I expect more work to be done in this area since multi-precision is so effective in increasing the accuracy of the inversion methods.

Chapter 2

The Noise Handling

Properties of the Talbot

Algorithm for Numerically

Inverting the Laplace

Transform

2.0.1 Introduction

This chapter examines the noise handling properties of three of the most widely used algorithms for numerically inverting the Laplace Transform. After examining the genesis of the algorithms, the regularization properties are evaluated through a series of standard test functions in which noise is added to the inverse transform. Comparisons are then made with the exact data. Our main

finding is that the Talbot inversion algorithm performs with greater accuracy when compared to the Fourier Series and Stehfest numerical inversion schemes, as they are outlined in this chapter.

(This chapter was published as a research paper: First published September 13, 2018, Research Article <https://doi.org/10.1177/1748301818797069>)

2.1 The Laplace Transform

The Laplace Transform is an integral transform defined as follows:

Let $f(t)$ be defined for $t > 0$, then the Laplace transform of $f(t)$ is given by:

$$\mathcal{L}\{f(t)\} = \int_0^{\infty} f(t)e^{-st} dt \quad (2.1)$$

with $\mathcal{L}\{f(t)\}$ denoted as $F(s)$. The Laplace transform can be shown to exist for any function which can be integrated over any finite interval $0 < t < l$ for $l > 0$, and for which $f(t)$ is of exponential order, i.e.

$$|f(t)| < Me^{at} \quad (2.2)$$

as $t \rightarrow \infty$, where M and a are small real positive numbers.

Analytically the inverse Laplace transform is usually obtained using the techniques of complex contour integration with the resulting set of standard transforms presented in tables [97].

However, using the Laplace Transform to obtain solutions of differential equations can lead to solutions in the Laplace domain which are not easily invertible to the real domain by analytical means. Thus numerical inversion techniques are used to convert the solution from the Laplace to the real domain.

2.2 The Inverse Laplace Transform Perturbation and Precision

The recovery of the function $f(t)$ is via the inverse Laplace transform which is most commonly defined by the Bromwich integral formula

$$\mathcal{L}^{-1}\{F(s)\} = f(t) = \frac{1}{2\pi i} \int_{u-i\infty}^{u+i\infty} F(s) e^{st} ds \quad (2.3)$$

for some $u \in \mathbb{R}$. [97]

The choice of s in (2.1) and so in (2.3) is not an arbitrary one. If s is chosen so that it lies on the positive real axis, the solution of (2.3) is being treated as a positive real integral equation. The problem here is that the inverse problem is known to be ill-posed, meaning that small changes in the values of $F(s)$ can lead to large errors in the values for $f(t)$ [10].

Hence when Laplace Transform methods are used in finding numerical solutions to partial differential equations, the corresponding inversion methods can be highly sensitive to the inevitable noisy data that arises in their computation via truncation and round off error, a process which is exacerbated in non-linear schemes. Abate and Valko [1] have shown that, to some extent, these errors can be curtailed by working in a multi-precision environment; as I show in the “Tests” section later, a small amount of noise in the data can significantly perturb the solution. When this is the case, it becomes difficult for unlimited precision to aid in the convergence of the algorithm to the correct solution.

2.3 The Algorithms

There are over 100 algorithms available for inverting the Laplace Transform with numerous comparative studies. Examples include Duffy [40], Narayanan and Beskos [78], Cohen [23], and perhaps the most comprehensive by Davies and Martin [36]. However for the purposes of this investigation, we apply our tests using “Those algorithms that have passed the test of time” [1], this is because these algorithms are reported to give the most accurate results on the widest variety of functions. [36],[40]. These fall into four groups,

- (1) Fourier series expansion.
- (2) Combination of Gaver Functionals.
- (3) Laguerre function expansion.
- (4) Deformation of the Bromwich contour.

Derivations of particular versions of these algorithms are given in the next section. In the upcoming sections, we examine the Stehfest algorithm, which is a widely used version of the Gaver functionals and Talbot Algorithm that uses a particular deformation of the Bromwich contour.

However, for now, we do not run our tests using the Laguerre function expansion. While we do intend to investigate this method later on in our work, our choices in this work have been made based on the ease of implementation of the inversion method, an issue connected to parameter choice and control. The Laguerre method requires more than two parameters to effectively compute the desired transform, while the other three methods can perform reasonably well when defined using just the one parameter.

2.3.1 The Fourier Series Method

In their survey of algorithms for inverting the Laplace Transform, Davies and Martin [36] note that the Fourier series method without accelerated convergence gives good accuracy on a wide variety of functions. Since the Laplace Transform is closely related to the Fourier transform it is not surprising that inversion methods based on a Fourier series expansion would yield accurate results. In fact, the two sided Laplace transform can be derived from the Fourier transform in the following way. We can define the Fourier transform as

$$\mathcal{F}\{f(t)\} = \int_{-\infty}^{\infty} f(t) e^{-2\pi i \nu t} dt \quad (2.4)$$

providing $f(t)$ is an absolutely integrable function, i.e.

$$\int_{-\infty}^{\infty} |f(t)| dt < \infty \quad (2.5)$$

Then letting $\nu = 2\pi i$ we have

$$\mathcal{F}\{f(t)\} = \int_{-\infty}^{\infty} f(t) e^{-i \nu t} dt \quad (2.6)$$

As many functions do not satisfy the condition in (2.5), $f(t)$ is multiplied by the exponential dampening factor e^{-ut} thus

$$\mathcal{F}\{f(t)e^{-ut}\} = \int_{-\infty}^{\infty} f(t) e^{-i \nu t} e^{-ut} dt \quad (2.7)$$

and letting $s = u + i \nu$ we obtain the two sided Laplace transform of $f(t)$ as

$$\mathcal{F}\{f(t)e^{-ut}\} = \mathcal{L}\{f(t)\} = \int_{-\infty}^{\infty} e^{-st} f(t) dt \quad (2.8)$$

LePage [14] noted that the integral in (2.8) can be written in two parts as

follows:

$$\int_{-\infty}^{\infty} e^{-st} f(t) dt = \int_{-\infty}^0 e^{-st} f(t) dt + \int_0^{\infty} e^{-st} f(t) dt \quad (2.9)$$

The second term in the above expression is referred to as the one-sided Laplace transform or simply the Laplace transform. Thus s is defined as a complex variable in the definition of the Laplace Transform.

As before the inverse Laplace transform is given as:

$$f(t) = \frac{1}{2\pi i} \int_{u-i\infty}^{u+i\infty} e^{st} F(s) ds \quad (2.10)$$

With $s = u + iv$ in (2.10) this leads to the result

$$f(t) = \frac{2e^{ut}}{\pi} \int_0^{\infty} [\operatorname{Re}\{F(u + iv)\} \cos(vt) - \operatorname{Im}\{F(u + iv)\} \sin(vt)] dv \quad (2.11)$$

As Crump [7] points out equations (2.1) and (2.3) can be replaced by the cosine transform pair

$$\operatorname{Re}\{F(u + iv)\} = \int_0^{\infty} e^{-ut} f(t) \cos(vt) dt \quad (2.12)$$

$$f(t) = \frac{2e^{ut}}{\pi} \int_0^{\infty} \operatorname{Re}\{F(u + iv)\} \cos(vt) dv \quad (2.13)$$

or by the sine transform pair

$$\operatorname{Im}\{F(u + iv)\} = - \int_0^{\infty} e^{-ut} f(t) \sin(vt) dt \quad (2.14)$$

$$f(t) = - \frac{2e^{ut}}{\pi} \int_0^{\infty} \operatorname{Im}\{F(u + iv)\} \sin(vt) dv \quad (2.15)$$

Dunbar and Abate [11] applied a trapezoid rule to (2.13) resulting in the Fourier

series approximation,

$$f(t) \approx \frac{2e^{ut}}{T} \left[\frac{1}{2} F(u) + \sum_{k=1}^{\infty} \operatorname{Re} \left\{ F\left(u + \frac{k\pi i}{T}\right) \right\} \cos\left(\frac{k\pi t}{T}\right) \right] \quad (2.16)$$

where $f(t)$ is expanded in the interval $0 \leq t < T$. For faster computation Simon and Stroot [20] proposed the following version of equation (2.16)

$$f(t) \approx \frac{e^{ut}}{t} \left[\frac{1}{2} F(u) + \sum_{k=1}^{\infty} \operatorname{Re} \left\{ F\left(u + \frac{k\pi i}{t}\right) \right\} (-1)^k \right] \quad (2.17)$$

This series can be summed much faster than (2.16) as there are no cosines to compute [25]. This algorithm is relatively easy to implement, with u being the only real varying parameter.

However as pointed out by Crump [28] for the the expression in (2.17) the transform $F(s)$ must now be computed for a different set of s - values for each distinct t . Since this type of application often occurs in practice in which the numerical computations of $F(s)$ is itself quite time consuming this may not be an economical inversion algorithm to use. These drawbacks to some extent, can be overcome by using the fast Fourier transform techniques [24], [25].

Crump [28] also extends this method to one of faster convergence by making use of the already computed imaginary parts. There are several other acceleration schemes, for example, those outlined by Cohen [23]; however, these acceleration methods, in general, require the introduction of new parameters, which for the purpose of this investigation, we wish to avoid.

2.3.2 The Stehfest Algorithm

Davies and Martin [36] cite the Stehfest [99] algorithm as providing accurate results on a variety of test functions. Since that time, this algorithm has become widely used for inverting the Laplace Transform, being favoured due to

its reported accuracy and ease of implementation.

Here we give a brief overview of the evolution of the algorithm from a probability distribution function to the Gaver functional, whose asymptotic expansion leads to an acceleration scheme which yields the algorithm in its most widely used form.

Gaver [50] investigated a method for obtaining numerical information on the time dependent behaviour of stochastic processes, which often arise in queueing theory. The investigation involved examining the properties of the three parameter class of density functions, namely

$$p_{n,m}(a;t) = \frac{(n+m)!}{n!(m-1)!} (1 - e^{-at})^n a e^{-mat} \quad (2.18)$$

with $n, m \in \mathbb{N}$. After the binomial expansion of the term $(1 - e^{-at})^n$, Gaver went on to find the expectation $E[f(T_{n,m})]$ where $T_{n,m}$ is the random variable with density (2.18). From this Gaver was able to express the inverse Laplace transform in terms of the functional

$$f_{n,m}(t) = \frac{\ln 2}{t} \frac{(n+m)!}{n!(m-1)!} \sum_{j=0}^n \binom{n}{k} (-1)^k F\left((k+m)\frac{\ln 2}{t}\right) \quad (2.19)$$

with certain conditions on n and m , Gaver makes $n = m$ and expresses equation (2.19) as

$$f_n(t) = \frac{\ln 2}{t} \frac{(2n)!}{n!(n-1)!} \sum_{k=0}^n \binom{n}{k} (-1)^k F\left((k+n)\frac{\ln 2}{t}\right) \quad (2.20)$$

While the expression in (2.20) can be used to successfully invert the Laplace transform for a large class of functions its rate of convergence is slow [35], [40]. However Gaver [13] has shown that (2.20), with $a = \frac{\ln 2}{t}$ has the asymptotic

expansion

$$f_n(t) \approx f\left(\frac{\ln 2}{a}\right) + \frac{\alpha_1}{n} + \frac{\alpha_2}{n^2} + \frac{\alpha_3}{n^3} + \dots \quad (2.21)$$

where the α_j 's are constant coefficients in the asymptotic series. Hence (2.21) converges to the limit

$$\lim_{n \rightarrow \infty} f_n(t) = f\left(\frac{\ln 2}{a}\right)$$

as $n \rightarrow \infty$. For the conditions on m and n and justification for the substitution for a referred to above, see Gaver [50]. This asymptotic expansion provides scope for applying various acceleration techniques enabling a more viable application of the basic algorithm.

Stehfest's acceleration scheme.

For the purposes of following Stehfest's derivation it is convenient to rewrite (2.20) as

$$f_n(t) = F_n = \frac{(2n)!a}{n!(n-1)!} \sum_{j=0}^n \binom{n}{j} (-1)^j F((k+n)a) \quad (2.22)$$

with $a = \frac{\ln 2}{t}$. Stehfest [21] begins by supposing we have N values for $F[(k+n)a]$ with $F(a)$, $F(2a)$, $F(3a)$, ..., $F(Na)$ for N even. Using equation (2.22) we can then determine $\frac{N}{2}$ values $F_1, F_2, \dots, F_{N/2}$. Now each of these $N/2$ values satisfy the asymptotic series in (2.21) with the same α_j coefficients.

As pointed out by Stehfest, the α_j 's are the same for each of the above expressions and by using a suitable linear combination we can eliminate the first $(\frac{N}{2} - 1)$ error terms in equation (2.21) can be eliminated. That is

$$f\left(\frac{\ln 2}{a}\right) = \sum_{n=1}^{\frac{N}{2}} a_n F_{(\frac{N}{2}+i-1)} + O\left(\frac{1}{N^{\frac{N}{2}}}\right) \quad (2.23)$$

which may be achieved by selecting the coefficients to satisfy

$$\sum_{n=1}^{\frac{N}{2}} a_n \frac{1}{(\frac{N}{2} + 1 - n)^k} = \delta_{k,0} \quad k = 1, \dots, N/2 - 1 \quad (2.24)$$

$$a_n = \frac{(-1)^{n-1}}{(\frac{N}{2})!} \binom{\frac{N}{2}}{n} n \left(\frac{N}{2} + 1 - n \right)^{\frac{N}{2}-1} \quad (2.25)$$

Finally, Stehfest substitutes (5.28) into (2.23) and obtains the inversion formula

$$f(t) \approx \frac{\ln 2}{t} \sum_{j=1}^N A_j F\left(\frac{j \ln 2}{t}\right) \quad (2.26)$$

where

for N even.

$$A_j = (-1)^{\frac{N}{2}+j} = \sum_{k=\lfloor \frac{j+1}{2} \rfloor}^{\min(j, \frac{N}{2})} \frac{k^{\frac{N}{2}} (2k)!}{(\frac{N}{2} - k)! k! (k-1)! (j-k)! (2k-j)!} \quad (2.27)$$

[99].

2.3.3 The Talbot Algorithm.

Equations (2.4) to (2.8) showed that the Laplace transform can be seen as a Fourier transform of the function

$$e^{-ut} f(t) \quad t > 0 \quad (2.28)$$

i.e.

$$\mathcal{F}\{e^{-ut} f(t)\} = \mathcal{L}\{f(t)\} = F(s) \quad (2.29)$$

Hence the Fourier transform inversion formula can be applied to recover the function thus

$$\mathcal{F}^{-1} \{F(s)\} = e^{-ut} f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(s) e^{ivt} dv \quad (2.30)$$

as $s = u + iv$ we have that $ds = idv$ and so

$$\frac{1}{2\pi i} \int_{-\infty}^{\infty} F(s) e^{st} ds \quad (2.31)$$

as $s = u + iv$ we have that $ds = idv$ and so

$$f(t) = \frac{1}{2\pi i} \int_{u-i\infty}^{u+i\infty} F(s) e^{st} ds \quad (2.32)$$

This result provides a direct means of obtaining the inverse Laplace transform.

In practice the integral in (2.32) is evaluated using contour integration

$$\frac{1}{2\pi i} \int_B e^{st} F(s) ds \quad (2.33)$$

with B denoting the Bromwich contour [98]. The contour is chosen so that it encloses all the possible singularities of $F(s)$. The idea of the contour is introduced so that the residue theorem can be used to evaluate the integral. However, when $f(t)$ is to be calculated using numerical quadrature, it may be more appropriate to devise a new contour. To ensure the convergence of (2.33) we may wish to control the growth of the magnitude of the integrand e^{st} by moving the contour to the left so, giving the real part of s a large negative component [1], [75].

However, the deformed contour must not be allowed to pass through any singularities of $F(s)$. This is to ensure that the transform is analytic in the region to the right of B .

Derivation of the Fixed Talbot Contour.

In the derivation that follows [1] and [75], are used as the primary basis for extending the explanation of the derivation of the Talbot algorithm for inverting the Laplace Transform.

Abate and Valko [1] begin with the Bromwich inversion integral along the Bromwich contour B with the substitution

$$F(s) = \frac{1}{s^\alpha}, \quad \alpha > 0 \quad (2.34)$$

So $f(t)$ can be expressed as

$$f(t) = \frac{1}{2\pi i} \int_B e^{t(s - a \log_e s)} ds \quad (2.35)$$

with $a = \frac{\alpha}{t}$ in (2.34) and (2.35). As pointed out by Abate and Valko [1] numerically evaluating the integral in (2.35) is difficult due to the oscillatory nature of the integrand.

However this evaluation can be achieved by deforming the contour B into a *path of constant phase*, thus eliminating the oscillations in the imaginary component. These paths of constant phase are also paths of steepest decent for the real part of the integrand [1],[11],[75].

There are in general a number of contours for which the imaginary component remains constant so we choose one on which the real part attains a maximum on the interior (a saddle point) and this occurs at $g'(s) = 0$ at some point on the contour. At these saddle points the $Im\{g(s)\} = 0$ [75]. Here

$$g(s) = s - a \ln s \quad (2.36)$$

in (2.35). Thus we have

$$g'(s) = 1 - \frac{a}{s} \quad (2.37)$$

So the stationary point occurs when $s = a$.

With $s = u + iv$ we have

$$\text{Im}\{u + iv - a \ln(u + iv)\} = 0 \quad (2.38)$$

Expressing $u + iv$ as $Re^{i\theta}$ we have

$$\text{Im}\{(u - a \ln R) + i(v - a\theta)\} = 0 \quad (2.39)$$

then

$$v = a\theta \quad (2.40)$$

and as

$$\tan(\theta) = \left(\frac{v}{u}\right) \quad (2.41)$$

then

$$u = a\theta \cot(\theta) \quad (2.42)$$

[1].

With $v = a\theta$ then s can be parametrized to Talbots contour:

$$s(\theta) = a\theta(\cot(\theta) + i) \quad -\pi < \theta < +\pi \quad (2.43)$$

[104].

Conformal mapping of the Talbot contour.

While the above parametrization can be used as a basis for inverting the Laplace Transform we proceed with the algorithm's development via a convenient con-

formal mapping as follows.

$$\cot \theta = \frac{i(e^{i\theta} + e^{-i\theta})}{(e^{i\theta} - e^{-i\theta})} \quad (2.44)$$

Then

$$\theta \cot \theta + i\theta = \frac{2i\theta}{1 - e^{-2i\theta}} \quad (2.45)$$

with $z = 2i\theta$ then (2.45)

$$= \frac{z}{1 - e^{-z}} \quad (2.46)$$

The function

$$S(z) = \frac{z}{1 - e^{-z}} \quad (2.47)$$

maps the closed interval $M = [-2\pi i, 2\pi i]$ on the imaginary z -plane onto the curve L in the s plane giving the integral,

$$f(t) = \frac{1}{2\pi i} \int_L F(s) e^{st} ds \quad (2.48)$$

For the details of this transformation, one can refer to the study of Logan [66].

Next we follow the procedure as adopted by Logan [66] for numerically integrating equation (2.48). With the change of variable (2.48) becomes

$$f(t) = \frac{1}{2\pi i} \int_M F(S(z)) e^{S(z)t} S'(z) dz \quad (2.49)$$

where

$$S'(z) = \frac{1 - (1 + z)e^{-z}}{(1 - e^{-z})^2} \quad (2.50)$$

For convenience we write,

$$f(t) = \frac{1}{2\pi i} \int_M Q(z) dz \quad (2.51)$$

where

$$Q(z) = F(s(z)) e^{s(z)t} s'(z) \quad (2.52)$$

and $M = [-2\pi, 2\pi]$. Then if we let $w = -iz$ for the integral in (2.51) so the interval of integration is now real and becomes $[-2\pi, 2\pi]$. Then using the trapezoid rule with n we obtain

$$f(t) \cong \frac{1}{n} \left\{ (I(2\pi i) + T(-2\pi i) + 2 \sum_{j=1}^{n-1} I(iw_j)) \right\} \quad (2.53)$$

where

$$w_j = 2\pi \left(\frac{2j}{n} - 1 \right) \quad (2.54)$$

and we note that $I(2\pi i) = I(-2\pi i) = 0$. [66].

The regularization properties of the Talbot algorithm

Despite the intricacies of deriving the Talbot algorithm, we have found it to be a relatively easy algorithm to implement. Also, the tests which we have carried out so far show that the algorithm performs with a high degree of accuracy. Moreover, the algorithm converges much faster than the Fourier series method without requiring the use of any acceleration schemes. Additionally, in the form in which we have used it there is only one parameter to control.

Perhaps its greatest strength is the fact that *we have found that it is able to handle noisy data (of magnitude outlined below) with little growth in the corresponding error*. As we will show, this is not the case for either the Fourier series or the Stehfest inversion algorithms presented above. Moreover, this "regularization property" does not exist for many of the numerical inversion schemes, as indicated by Egonmwan [43]. For most algorithms, this is generally overcome by constructing some regularization scheme which then needs to be attached

on to the inversion algorithm(s) of choice. This, of course, increases the complexity of the inversion process involving new parameters, thus requiring even greater knowledge of the desired solution. This is even more so if the scheme also involves some additional accelerated convergence process. For the Talbot scheme, one needs only to directly apply the algorithm, which has the ability to mitigate this level of noise. On the other hand, for the Stehfest method, a regularization scheme such as the Tikhonov regularisation scheme implemented by Egonmwan may need to be constructed for this purpose, and so some estimate of the noise will be required. In its simplest form, this process involves adding positive elements to the diagonals of the ill-conditioned matrix in order to decrease its condition number. This is unnecessary for the Talbot scheme as these ill-conditioned matrices do not exist. In a sense, then, the Talbot scheme has an implicit regularisation scheme when compared to the Stehfest algorithm.

As we pointed out earlier, the perturbation in the numerical schemes are a consequence of the inversion being carried out in the complex plane. The inclusion of complex arithmetic in the Talbot scheme enormously diminishes this perturbation as it does not create ill conditioned matrices associated with inversion schemes which are not done in the complex plane, [10]. Of great importance here too is that the *“regularization properties ” of the Talbot algorithm means that very good performance can be obtained on many of the test functions without the necessity for multi-precision.*

Egonmwan [43] examines regularised and collocation methods for the numerical inversion of the Laplace transform, which involve a Tikhonov regularisation scheme [106] based methods. This is then applied to the Stehfest [99] and Piessens [84] methods on various standard test functions for both exact $F(s)$ and noisy $F(s + \epsilon)$ data, where ϵ denotes the level of noise added.

For the Stehfest [99], Piessens [84] and the regularized method Egonwan [43] added noise of a magnitude $10^{-3} \times U(1, 0)$ to the inverse transforms in table 2.1, where $U(1, 0)$ is a random number between 1 and 0, (Uniform distribution) to the Laplace transform values. Commenting on his results, Egonwan notes "the Gaver Stehfest method gave very nice approximate solutions for a wide range of functions. However, it completely failed in the presence of noisy data. In the case of exact data, the method produced better numerical approximations when compared to the Piessins and the regularized collocation methods. However, the Piessins method gave better results than the regularized collocation method in the case of exact data."

In other words, methods which performed well for exact data did not do well for noisy data, and the regularized collocation method failed (as outlined by Egonmwan [43]) for exact data. Thus to use such regularized methods requires some a priori knowledge of the magnitude of the noise involved and, by implication, a better estimation of the solution than might be otherwise possible.

2.3.4 Tests

Table 2.1 lists the functions together with a variety of properties for the purpose of testing the noise handling capability of the three inversion algorithms employed.

No.	$F(s)$	$f(t)$	Function type
1	$\frac{s}{(s^2+1)^2}$	$0.5t \sin(t)$	Oscillating increasing
2	$\frac{1}{(s+1)^2}$	te^{-t}	Exponentially decreasing
3	$\frac{1}{s^5}$	$\frac{1}{24}t^4$	Increasing
4	$\frac{1}{\sqrt{s}}$	$\frac{1}{\pi t}$	With singularities
5	$\text{erf}\{\frac{2}{\sqrt{s}}\}$	$\frac{1}{\pi t} \sin(4\sqrt{t})$	Oscillating with singularities
6	$\frac{1}{s^2-0.5^2}$	$\sinh(0.5t)$	Hyperbolic
7	$\frac{s^3}{s^4+4(0.5)^4}$	$\cos(0.5t) \cosh(0.5t)$	Combination of oscillating and hyperbolic
	$\frac{\ln s}{s}$	$-(\ln t + \gamma)$	Natural log

Table 2.1: Test Functions

These functions are the same used by Egonmwan [43]. This sample of test functions has a variety of properties which we think form a basis for testing the robustness of the inversion schemes. We use three error measures, the L_2 norm defined as

$$E_2 = \sqrt{\sum_{i=1}^{40} \left| f_{\text{numerical}}(t_i) - f_{\text{exact}}(t_i) \right|^2}, \quad i = 1..40 \quad (2.55)$$

the L_∞ norm as

$$E_\infty = \max \left| f_{\text{numerical}}(t_i) - f_{\text{exact}}(t_i) \right|, \quad i = 1..40 \quad (2.56)$$

and the percentage error as

$$E_{\text{max}} = \max \left| \frac{f_{\text{numerical}}(t_i) - f_{\text{exact}}(t_i)}{f_{\text{exact}}(t_i)} \times 100 \right|, \quad i = 1..40 \quad (2.57)$$

To give a good estimation of the errors involved we have sampled t over 40 points for $t = 0.1$ to 4. The L_2 norm is chosen as a measure which averages out the error over the sample points, while the L_∞ norm and the % error as defined above choose the maximum error obtained for these measures. In all cases, the magnitude of noise added is $10^{-3} \times U(1, 0)$, where U is the uniform distribution.

The precision used for implementing the three algorithms is $1.8M$, where M is the number of weights for the Stehfest algorithm and $2M$ where M is the number of terms in the summation for the Talbot and the Fourier methods. The choice of these levels of precision is based on trial and error for the best performance of these methods.

They are perhaps larger than they need to be, but as our interest in this investigation is not on their efficiency but on their ability to handle noisy data, we wanted to ensure that the precision played as little part as possible in assessing their performance. Thus in cases where the extended precision decreases the accuracy of the noisy data, we used the usual double precision for these inversions.

For functions which have sine, cosine and hyperbolic properties, we increase the weights for the Stehfest. This is because these functions require more weights and a corresponding increase in precision for the Stehfest method to produce

accurate results. For the Fourier Series method we choose the parameter value of $a = 4$. Once again this choice is based on trial and error. We have found that this choice for a gives the best results for inverting the widest class of functions.

2.3.5 Results

		No Noise			Noise		
Method	M	L_2	L_∞	%error	L^2	L_∞	%error
Stehfest	30	9.4(-4)	5.0(-4)	3.8(-2)	4.6(16)	3.6(16)	1.2(18)
Talbot	55	2.0(-6)	5.4(-7)	2.3(-4)	6.2(-4)	2.7(-4)	3.7(-2)
Fourier	55	4.2(-2)	1,8(-3)	3.1(-1)	8.9(1)	2.9(0)	1.1(3)

Table 2.2: $f(t) = 0.5t \sin(t) = L^{-1}\{\frac{s}{(s^2+1)^2}\}$

		No Noise			Noise		
Method	M	L_2	L_∞	%error	L_2	L_∞	%error
Stehfest	16	1.1(-4)	4.0(-5)	5.4(-1)	3.0(7)	2.4(7)	2.6(10)
Talbot	55	7.3(-6)	6.4(-6)	2.1(-3)	7.8(-4)	2.3(-4)	3.1(-1)
Fourier	55	3.6(-3)	1.0(-2)	4.9(-0)	1.1(0)	9.0(-1)	9.7(2)

Table 2.3: $f(t) = te^{-t} = L^{-1}\{\frac{1}{(s+1)^2}\}$

		No Noise			Noise		
Method	M	L_2	L_∞	%error	L^2	L_∞	%error
Stehfest	16	6.7(-6)	3.0(-54)	2.8(-3)	3.8(3)	2.4(3)	1.1(12)
Talbot	55	3.8(-10)	3.4(-10)	5.1(-4)	2.3(-3)	8.8(-4)	1.5(-1)
Fourier	55	6.2(-1)	2.9(-1)	2.7(0)	7.6(0)	16.3(1)	2.5(3)

Table 2.4: $f(t) = \frac{1}{24}t^4 = L^{-1}\{\frac{1}{(s)^5}\}$

		No Noise			Noise		
Method	M	L_2	L_∞	%error	L_2	L_∞	%error
Stehfest	16	2.7(-8)	1.3(-8)	7.2(-7)	1.5(7)	1.2(7)	6.5(8)
Talbot	55	9.2(-2)	9.2(-3)	5.2(-2)	9.2(-2)	9.2(-3)	5.2(-2)
Fourier	55	6.2(-1)	2.9(-1)	2.7(0)	1.4(1)	6.3(0)	7.1(6)

Table 2.5: $f(t) = \frac{1}{\sqrt{\pi t}} = L^{-1}\{\frac{1}{(\sqrt{s})}\}$

		No Noise			Noise		
Method	M	L_2	L_∞	%error	L_2	L_∞	%error
Stehfest	16	2.6(-4)	1.6(-4)	6.6(-1)	1.2(7)	9.6(6)	7.2(9)
Talbot	55	2.2(-2)	2.2(-2)	7.1(-1)	2.2(-1)	2.2(-2)	7.1(-1)
Fourier	55	1.8(1)	1.1(1)	4.3(3)	3.9(3)	2.2(3)	4.1(6)

Table 2.6: $f(t) = \frac{1}{\pi t} \sin(4\sqrt{t}) = L^{-1}\{erf(\frac{2}{\sqrt{s}})\}$

		No Noise			Noise		
Method	M	L_2	L_∞	%error	L_2	L_∞	%error
Stehfest	36	9.8(-3)	9.2(-3)	2.1(-5)	2.6(7)	2.0(7)	7.0(6)
Talbot	55	7.2(-6)	7.2(-6)	4.6(-6)	4.5(-4)	3.1(-4)	7.6(-3)
Fourier	55	1.4(-1)	1.4(-1)	1.9(0)	1.7(1)	5.8(0)	3.4(2)

Table 2.7: $f(t) = \frac{\sinh(0.5t)}{0.5} = L^{-1}\left\{\frac{1}{s^2-0.5^2}\right\}$

		No Noise			Noise		
Method	M	L_2	L_∞	%error	L^2	L_∞	%error
Stehfest	36/16	3.7(-4)	3.0(-4)	3.0(-4)	3.1(6)	2.4(6)	1.0(8)
Talbot	55	5.8(-4)	5.8(-4)	5.8(-1)	7.0(-4)	6.0(-4)	6.0(-2)
Fourier	55	9.4(-2)	6.0(-2)	3.5(-1)	9.0(1)	2.8(1)	5.2(4)

Table 2.8: $f(t) = \cosh(0.5t) \cos(0.5t) = L^{-1} \left\{ \frac{s^3}{s^4 + 0.5^2} \right\}$

		No Noise			Noise		
Method	M	L_2	L_∞	%error	L^2	L_∞	%error
Stehfest	16	1.9(-8)	1.2(-7)	2.8(-5)	1.4(7)	1.8(7)	2.4(9)
Talbot	55	6.9(-3)	6.9(-3)	4.0(-1)	7.1(-3)	7.1(-3)	4.1(-1)
Fourier	55	8.6(-1)	8.3(-2)	4.0(3)	1.2(2)	3.8(2)	6.3(3)

Table 2.9: $f(t) = -(\ln(t) + \gamma) = L^{-1} \left\{ \frac{\ln s}{s} \right\}$

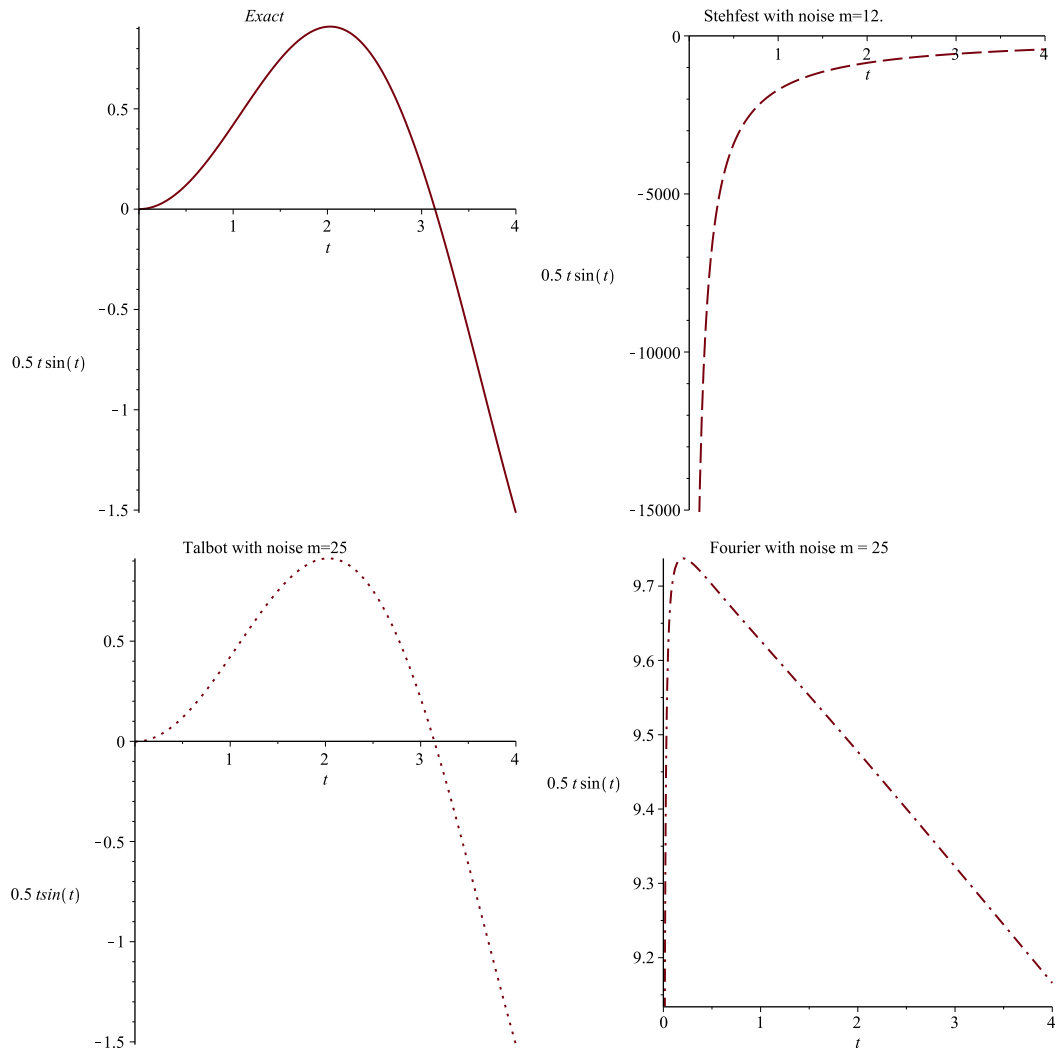


Figure 2.1: Numerical Reconstruction of $f(t) = 0.5t \cdot \sin(t) = L^{-1}\left\{\frac{s}{(s^2+1)^2}\right\}$

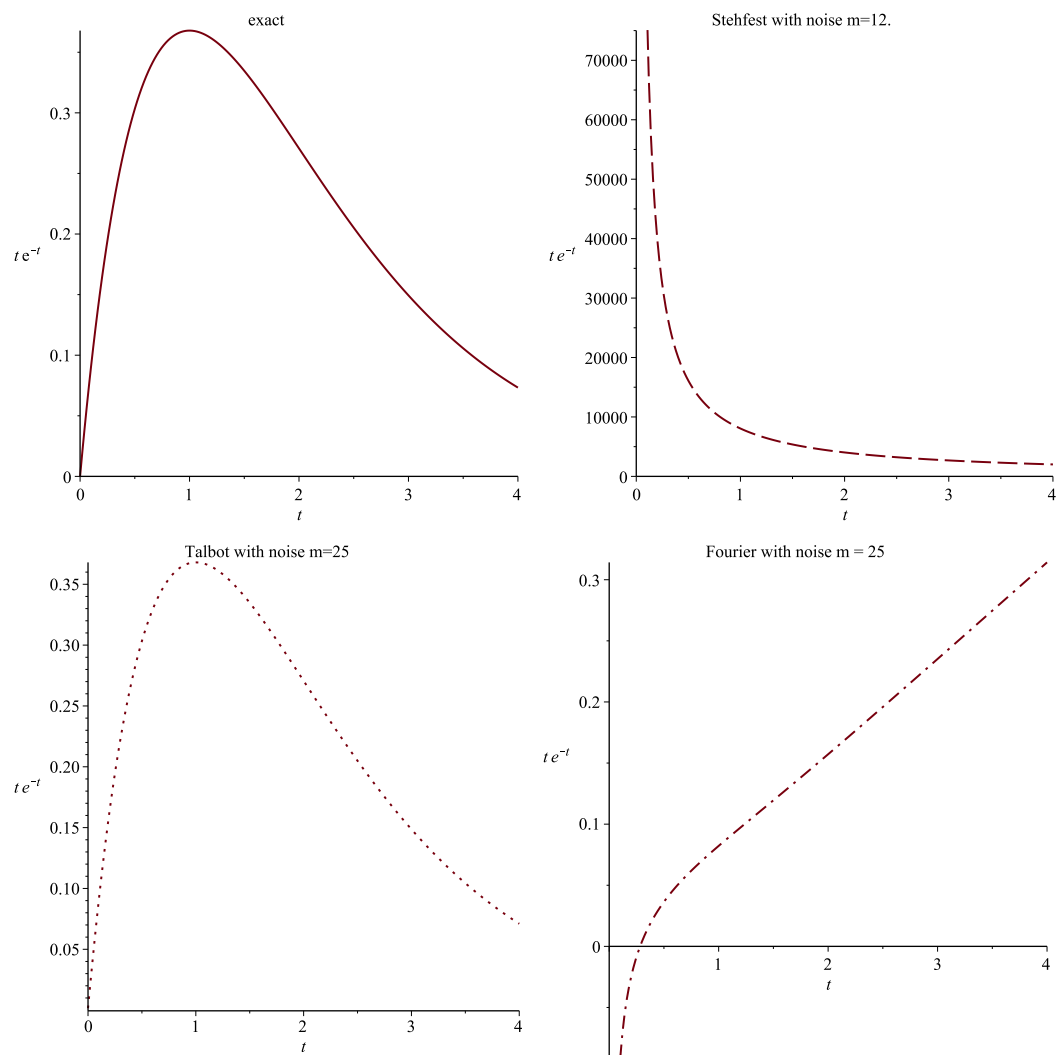


Figure 2.2: Numerical Reconstruction of $f(t) = te^{-t} = L^{-1}\{\frac{1}{(s+1)^2}\}$

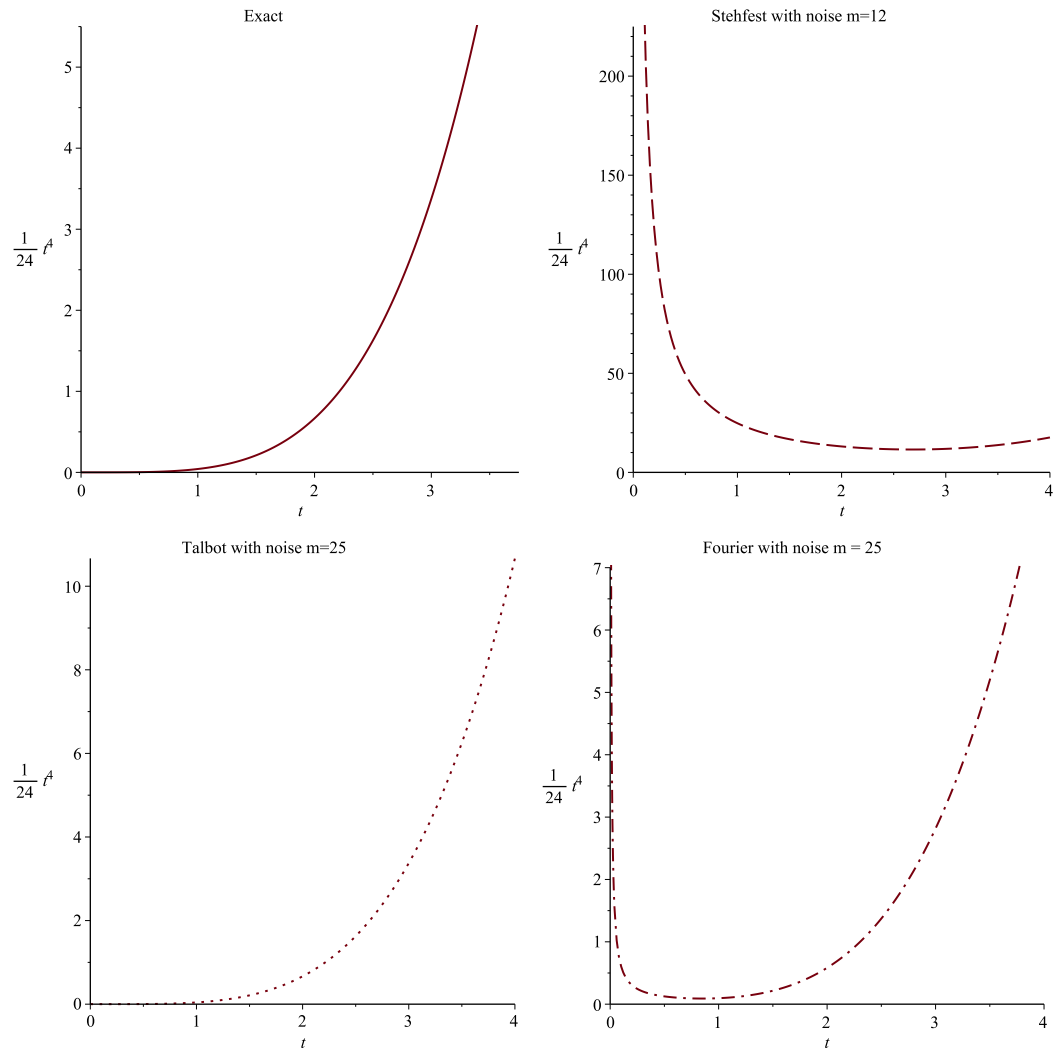


Figure 2.3: Numerical Reconstruction of $f(t) = \frac{1}{24}t^4 = L^{-1}\{\frac{1}{s^5}\}$

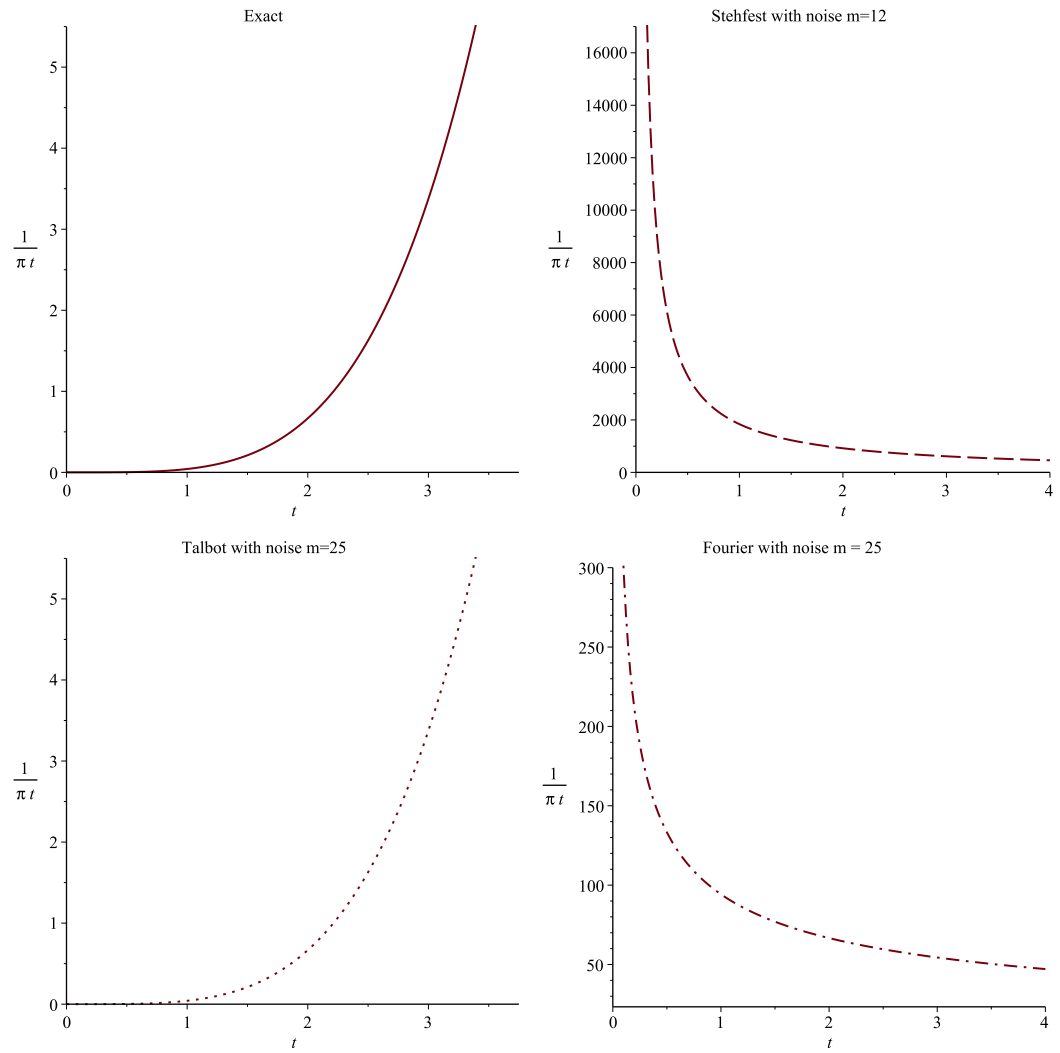


Figure 2.4: Numerical Reconstruction of $f(t) = \frac{1}{\pi t} = L^{-1}\left\{\frac{1}{\sqrt{s}}\right\}$

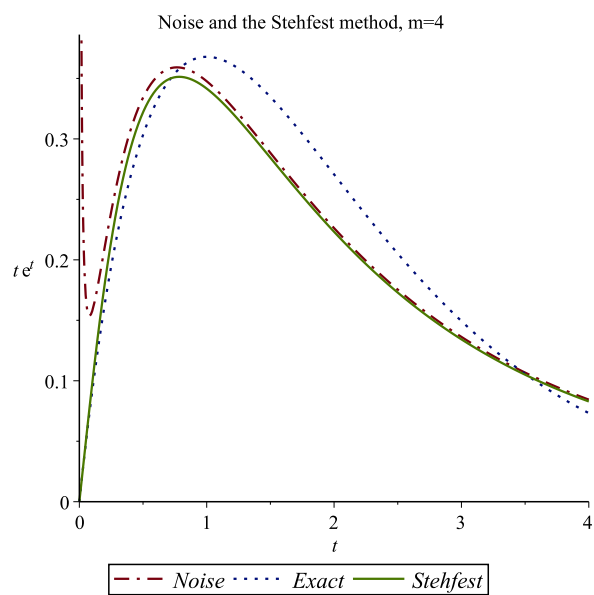


Figure 2.5: Numerical Reconstruction of $f(t) = 0.5t \cdot \sin(t) = L^{-1}\left\{\frac{s}{(s^2+1)^2}\right\}$

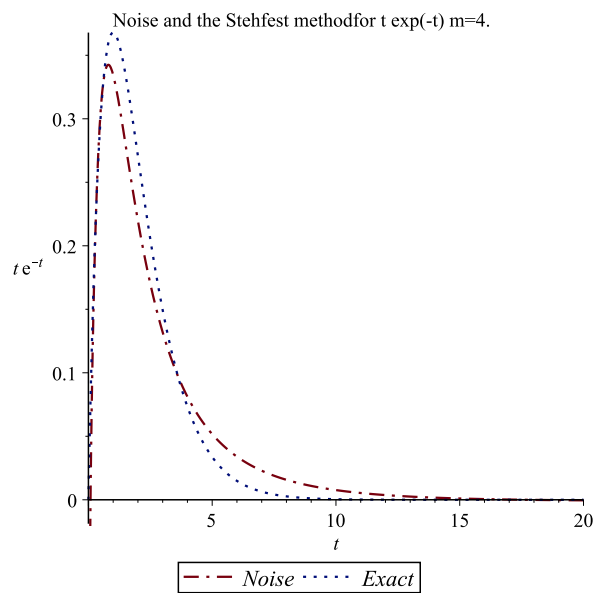


Figure 2.6: Numerical Reconstruction of $f(t) = 0.5t \cdot \sin(t) = L^{-1}\{\frac{s}{(s^2+1)^2}\}$

Tables 2.2-2.9 and Figures 2.1 to 2.4 show very good performance of the Talbot algorithm in handling noisy data. (For brevity, we have included only four graphical results for the eight functions using different weights as the performance of these functions with a higher number of weights is well illustrated in the tables).

With the exception of the function $f(t) = \frac{1}{\pi t}$ in Table 2.5 (for which the L_2 norm and L_∞ norm maintain their very small size) the error for the Talbot inversion diminishes considerably as a function of M . However, for both the Fourier series and the Stehfest inversion methods, both measures of error increase as M increases.

In Table 2.6, we also observe that the $erf(\frac{2}{\sqrt{s}})$ performs badly for the Fourier series method in both the noisy and noise free environment. Table 2.8 includes two sets of weights for the Stehfest inversion algorithm. For the accurate inversion of sinusoidal functions, this algorithm requires more weights for increasing values of t ; here, for example, we use 36 weights. However, when noise is added, the accuracy decreases with the number of weights used; thus, in this case, for better performance, we have used 16 weights.

Table 2.9 again shows minimal error involved for the Talbot inversion when noise is added. Figure 2.5 and Figure 2.6 demonstrate that the Stehfest algorithm handles noisy data more accurately by decreasing the number of weights used. This is because the error generated in reconstructing the function from noisy data increases as the number of weights used rises. However the accuracy achieved by decreasing the number of weight is not sufficient to justify such an approach for handling noisy data. Moreover as we have stated a larger number of weights and the corresponding increase in precision is necessary for handling trigonometric and hyperbolic functions. We again note that no such considerations are necessary when employing the Talbot algorithm.

2.4 Summary.

This chapter investigated the noise handling properties of three of the most widely used algorithms for numerically inverting the Laplace transform. This aids in understanding the perturbed nature of these algorithms and helps determine which is affected least by perturbation. This is important since, in applying these inversion schemes to solving diffusion problems, understanding how they perform when inevitable errors are introduced by truncation and round off error can be critical.

The results show that the Talbot algorithm handles the noisy data extremely well, having very little impact on the final outcome. Both the Stehfest and the Fourier series methods fail to handle the noise. This is because rounding errors worsen as the number of weights used increases. This is due to the fact that a significant part of the perturbation in these numerical schemes is a consequence of the inversion being carried out on the real axis in the complex plane. The inclusion of complex arithmetic in the Talbot scheme enormously diminishes this perturbation. This has implications for implementing the LTFDM when solving nonlinear diffusion or time dependent parabolic partial differential equations, which can generate noisy data through a combination of measurement, truncation and round off error. Using the Talbot algorithm in these circumstances avoids additional complications such as having to devise regularized collocation methods to attain accurate solutions to these problems.

Having examined the noise handling properties of the algorithms, in the next chapter, the performance of five algorithms for numerically inverting the Laplace transform are investigated in standard double precision and multi-precision. This investigation will explain how these algorithms handle their perturbations in these precision environments. The knowledge gained can help make an informed choice of the best algorithm to use in double and multi-precision.

Chapter 3

The Numerical Inversion of the Laplace Transform in a Multi-Precision .

3.1 Introduction

This paper examines the performance of five algorithms for numerically inverting the Laplace transform in standard 16-digit and multi-precision environments. The algorithms, whose derivations are outlined in Section 4, are taken from three of the four main classes of numerical methods used to invert the Laplace transform [1].

The Abate-Valko [1] and Logan schemes [66] belong to the class of inversion algorithms which deform the Bromwich contour [98]. They are closely related versions of this approach as they both use Talbot’s method for deformation of the contour [104]. Logan, however, chooses an exponential transform while Abate-Valko extends the original Talbot formulation expressing the contour in trigonometric form.

The Stehfest and Salzer-Gaver algorithms [99], are again two closely related schemes based on the acceleration of the Gaver functional [50]. Stehfest applied a modified Salzer acceleration scheme [112] onto the Gaver functional simplifying this result to yield one of the most widely used algorithms for inverting the Laplace transform. We find, however, that when we implement a direct application of the Salzer acceleration scheme onto the Gaver functional (Salzer-Gaver) with Stehfest’s modifications, we do not obtain the same results as those produced by the Stehfest scheme. We conclude that Stehfest’s simplification process is at least in part responsible for the differences in performance of these two versions.

Finally, we examine the Fourier series method [28], which expresses the inversion integral as a Fourier series and then uses the trapezium rule to evaluate the integral numerically. The Fourier series method differs from the other four algorithms as no acceleration scheme is used to force convergence. This means that the series is not truncated, and therefore precision environments do not

affect the accuracy of the inversion process. The algorithm is only used in a standard 16 digit precision environment and is compared with the four other schemes using standard precision.

(This chapter was first published as a paper on May 23 2022 <https://doi.org/10.4236/am.2022.135027>)

(Section 2 and the first paragraph of Section 3 are taken from our earlier work [38] which is necessary to set the background for the rest of the paper).

3.2 The Laplace transform

The Laplace transform is an integral transform defined as follows:

Let $f(t)$ be defined for $t \geq 0$, then the Laplace transform of $f(t)$ is given by,

$$\mathcal{L}\{f(t)\} = \int_0^{\infty} f(t)e^{-st} dt \quad (3.1)$$

Thus $\mathcal{L}\{f(t)\}$ is a function of s denoted as $F(s)$. The Laplace transform can be shown to exist for any function $f(t)$, which can be integrated over any finite interval $0 < t < l$ for $l > 0$, and for which $f(t)$ is of exponential order, i.e.

$$|f(t)| < Me^{at} \quad (3.2)$$

as $t \rightarrow \infty$, where $M > 0$ is a finite real number and a is a small real positive number.

Analytically the inverse Laplace transform is usually obtained using the techniques of complex contour integration with the resulting set of standard transforms presented in tables [32].

However, using the Laplace transform can generate data in the Laplace domain which is not easily invertible to the real domain by analytical means. Thus numerical inversion techniques have to be used to convert the data from the

s -space to the time domain [38].

3.3 The Inverse Laplace Transform, Perturbation and Multi-precision

The recovery of the function $f(t)$ is via the inverse Laplace transform which is most commonly defined via the Bromwich contour integral,

$$L^{-1}\{F(s)\} = f(t) = \frac{1}{2\pi i} \int_{\alpha-i\infty}^{\alpha+i\infty} f(s) e^{st} ds \quad (3.3)$$

such that $\alpha \in R$. The inversion integral is widely known to be highly perturbed [23],[45],[60]. This is due to the e^{st} term in the integral, which amplifies small changes in the input data. Hence all numerical schemes are vulnerable to this perturbation, and this has to be taken into account when using the various algorithms to invert the Laplace transform.

As Abate-Valko notes [1], “In the traditional development of the inversion methods, most of the effort was directed at controlling round-off errors. This is because the process is numerically unstable in a fixed-precision computing environment. The problem is that as the user tries to increase the accuracy, there comes the point where the round off error causes the error to increase dramatically”.

In fact, Abate-Valko goes further and makes the claim that “for our purposes, we add the proviso that values of the transform can be computed to any desired precision as a function of the complex variable”.

This suggests that working in a multi-precision environment can act to reduce errors and the resulting perturbations which exist in transforming the data from the s -space into the time domain.

3.4 The Algorithms

We examine five algorithms drawn from three of the four main classes of algorithms for numerically inverting the Laplace Transform [1]. These three are the Fourier series expansion, methods which use the Gaver functional and deformation of the Bromwich contour. Here we give a brief outline of five algorithms drawn from these three classes. For a more extensive treatment of the derivation of these algorithms, please see our earlier work [38].

3.4.1 The Fourier Series Method

In their wide-ranging survey of Numerical Inversion of the Laplace Transform, Davies and Martin [36] cite the Fourier series approach without accelerated convergence as giving “good accuracy on a fairly wide range of functions”. (For a full treatment of the derivation of this algorithm please see [38]).

By letting $s = u + iv$ in (3.3) we can express the resulting cosine transfer pair as,

$$f(t) = \frac{2e^{ut}}{\pi} \int_0^\infty \operatorname{Re}\{F(u + iv)\} \cos(vt) dv \quad (3.4)$$

Simon et al. [94] then apply the trapezium rule to the expression in (3.4) and derive the expression

$$f(t) \approx \frac{2e^{ut}}{T} \left[\frac{1}{2} F(u) + \sum_{k=1}^{\infty} \operatorname{Re}\left\{F\left(u + \frac{k\pi i}{T}\right)\right\} \cos\left(\frac{k\pi t}{T}\right) \right] \quad (3.5)$$

where $f(t)$ is expanded in the interval $0 \leq t < T$. [39].

3.4.2 Gaver's Functional

Gaver [50] derived the function,

$$f_n(t) = \frac{\ln 2}{t} \frac{(2n)!}{n!(n-1)!} \sum_{k=0}^n \binom{n}{k} (-1)^k F\left((k+n)\frac{\ln 2}{t}\right) \quad (3.6)$$

for numerically inverting the Laplace transform. The problem with (3.6) is its slow convergence. However Gaver was also able to show that (3.6) has the asymptotic expansion

$$f_n(t) \approx f\left(\frac{\ln 2}{a}\right) + \frac{\alpha_1}{n} + \frac{\alpha_2}{n^2} + \frac{\alpha_3}{n^3} + \dots \quad (3.7)$$

where the α_j 's are constant coefficients in the asymptotic series. Hence (3.7) in the limit converges to

$$\lim_{n \rightarrow \infty} f_n(t) = f\left(\frac{\ln 2}{a}\right)$$

This means that it is possible to accelerate the convergence rate of (3.6). Much of the literature alludes to the fact that a Salzer [112] acceleration scheme is used on the Gaver functional in (3.6), which results in the Stehfest algorithm. In fact, Stehfest's approach was a little more subtle than a direct application of the Salzer acceleration.

Using Salzer acceleration

The Salzer acceleration scheme makes use of the "Toeplitz limit theorem" [112], "this concerns the convergence of a transformation of a sequence ζ_s where the $(n+1)$ th member of the transformed sequence is a weighted mean of the first $(n+1)$ terms"

$$\bar{S}_n = \sum_{k=0}^n \mu_{nk} \cdot S_k \quad (3.8)$$

Here $\overline{S}_n$ is the transformed sequence and S_k the original sequence and,

$$\mu_{nk} = (-1)^{n+k} \frac{(1+k)^n}{n!} \binom{n}{k} \quad (3.9)$$

For the sake of compatibility with (3.9) we make the change $k \rightarrow i$ and $n \rightarrow k$ in (3.6). With this change of variables we also write

$$\frac{(2k)!}{k!(k-1)!} = \frac{k(2k)!}{(k!k!)}$$

This allows the sum to be taken from $k = 0$ to n without $(0-1)!$ in the denominator of (3.6). So with Salzer acceleration we can express the Salzer-Gaver algorithm as

$$f_n(t) = \frac{\ln 2}{t} \sum_{k=0}^n (-1)^{n+k} \frac{(k+1)^n}{k!(n-k)!} \frac{k(2k)!}{k!k!} \sum_{i=0}^k \frac{k!}{i!(k-i)!} (-1)^i F\left\{ \frac{(k+i) \ln 2}{t} \right\} \quad (3.10)$$

Stehfest's acceleration scheme

For the purposes of following Stehfest's derivation it will be convenient to rewrite (3.6) as

$$f_n(t) = F_n = \frac{(2n)!a}{n!(n-1)!} \sum_{k=0}^n \binom{n}{k} (-1)^k F((k+n)a) \quad (3.11)$$

Stehfest [99] begins by supposing we have N values for $F[(k+n)a]$ with $F(a)$, $F(2a)$, $F(3a)$, ..., $F(Na)$ for N even. Using (3.11) we can then determine $\frac{N}{2}$ values $F_1, F_2, \dots, F_{N/2}$. Now each of these $\frac{N}{2}$ values satisfy the asymptotic series in (3.7) with the same coefficients.

As Stehfest [99] points out, the α_j 's in (3.7) are the same for each of the above expansions and by using a suitable linear combination the first $(\frac{N}{2} - 1)$ error terms in (3.7) can be eliminated. That is

$$f\left(\frac{\ln 2}{a}\right) = \sum_{n=1}^{\frac{N}{2}} a_n F_{(\frac{n}{2}+i-1)} + O\left(\frac{1}{N^{\frac{N}{2}}}\right) \quad (3.12)$$

which may be achieved by selecting the coefficients to satisfy

$$\sum_{n=1}^{\frac{N}{2}} a_n \frac{1}{(\frac{N}{2} + 1 - n)^k} = \delta_{k,0} \quad k = 1, \dots, N/2 - 1 \quad (3.13)$$

which produce the same coefficients as the Salzer acceleration scheme used in (3.8). In fact for any n , Stehfest generates the required coefficients using what is in effect a modified Salzer acceleration scheme giving

$$a_n = \frac{(-1)^{n-1}}{(\frac{N}{2})!} \binom{\frac{N}{2}}{n} n \left\{ \left(\frac{N}{2} + 1 - n \right)^{\frac{N}{2}-1} \right\} \quad (3.14)$$

Finally, Stehfest substitutes these results into (3.13) and gets the inversion formula

$$f(t) \approx \frac{\ln 2}{t} \sum_{j=1}^N A_j F\left(\frac{j \ln 2}{t}\right) \quad (3.15)$$

for N even and

$$A_j = (-1)^{\frac{N}{2}+j} = \sum_{k=\lfloor \frac{j+1}{2} \rfloor}^{\min(j, \frac{N}{2})} \frac{k^{\frac{N}{2}} (2k)!}{(\frac{N}{2} - k)! k! (k-1)! (j-k)! (2k-j)!} \quad (3.16)$$

However, a direct application of the modified Salzer acceleration scheme in (3.14) onto the Gaver functional in (3.11) does not produce the same results for the expression in (3.16) so they are not exactly equal to each other.

To show this we consider the function $\sin(t)$ whose Laplace transform is

$$\frac{1}{s^2 + 1}$$

The eight weights produced by the Salzer acceleration for $n = 8$ are exactly the same for $n = 18$ in Stehfest's modified Salzer acceleration scheme in (3.14). However, Table 4.1 shows that for $\sin(t)$ with these same weights, the Salzer-Gaver scheme produces different results when compared to Stehfest's scheme in (3.16). This is due to Stehfest's simplification of the Salzer-Gaver scheme to the expression in (3.16).

This simplification was necessary because, as we show in our results in Section 5, Stehfest's final expression in (3.16) is faster and works better in standard double precision. As the algorithm was developed in 1970, this would be far more efficient when taking into consideration the computing power available at the time. Again as we show in Section 5, a direct application of a Salzer acceleration scheme onto the Gaver functional is only advantageous in a multi-precision environment.

t	Stehfest	Salzer-Gaver
5	0.89	1.02
10	0.08	0.18
15	0.002	0.03
20	0.03	0.02
25	0.001	0.004
30	0.001	0.004

Table 3.1: Salzer-Gaver Stehfest for $\sin(t)$

3.4.3 Logan's Version Of The Talbot Algorithm

Logan begins with the transformation

$$s(z) = \frac{az}{1 - e^{-z}} \quad (3.17)$$

(For the details of this transformation one can refer the study of Logan [66].)

(with $z \in \mathbb{C}$) and constructs the integral

$$f(t) = \frac{1}{2\pi i} \int_M Q(z) \, dz \quad (3.18)$$

where

$$Q(z) = F(s(z)) e^{s(z)t} s'(z) \quad (3.19)$$

and $M - [2\pi i, 2\pi i]$. Then if we let $w = -iz$ for the integral in (3.18) so the interval of integration is now real and becomes $[-2\pi, 2\pi]$. Then using the trapezoid rule with n we obtain

$$f(t) \cong \frac{1}{n} \left\{ (I(2\pi i) + T(-2\pi i) + 2 \sum_{j=1}^{n-1} I(iw_j)) \right\} \quad (3.20)$$

where

$$w_j = 2\pi \left(\frac{2j}{n} - 1 \right) \quad (3.21)$$

and we note that $I(2\pi i) = I(-2\pi i) = 0$ [66], [38].

Abate and Valko's Version of The Talbot Algorithm

Abate and Valko [1] deform the Bromwich contour using the Talbot path which has the form,

$$s(h) = rh(\cot(h) + i), \quad -\pi < h < \pi \quad (3.22)$$

So we have

$$s'(h) = ir(1 + ir(h)) \quad (3.23)$$

where,

$$r(h) = h + (h \cot(h) - 1) \cot(h) \quad (3.24)$$

Then from (3.18) we find,

$$f(t) = \frac{r}{p} \int_0^P \operatorname{Re}[e^{ts(h)} F(s(h)(1 + ir(h)))] dh \quad (3.25)$$

They then approximate the value of the integral in (3.18) by using the trape-

zoidal rule with step size $\frac{p}{m}$ and $h_k = \frac{kp}{m}$ to get,

$$f(t, M) = \frac{r}{m} \left[\frac{1}{2} F(r) \exp(rt) + \sum_{k=1}^{M-1} \operatorname{Re}[e^{ts(h_k)} F(s(h_k))(1 + ir(h_k))] \right] \quad (3.26)$$

Based on numerical experiments, Abate-Valko then fix the parameter r to the value,

$$r = \frac{2M}{5t} \quad (3.27)$$

[1]. We also use this value for a in Logan's transformation.

3.5 Results

We tested the five algorithms on the functions listed in Tables 4.2 and 4.3 on pages 103 and 104. Functions 1-11 and 18 are taken from the 16 functions tested by Davies and Martin [36]. The remaining functions are selected from those tested by Abate-Valko [1].

The first set of tests was carried out using 16 digits double precision. These results are shown in Table 4.4. The Fourier, Logan and Abate-Valko schemes were run with weights $M = 50$, $M = 100$ and $M = 200$; however, for brevity, we include only the result for $M = 200$.

For the Stehfest and the Salzer-Gaver algorithms, the best results were obtained with weights of $M = 16$ and $M = 8$, respectively. This is in keeping with Stehfest's observations on the instability of this method as M increases above an optimal level [99].

In multi-precision, the number of precision digits for Abate-Valko was set equal to N [1], and for the Slazer-Gaver and Stehfest schemes, best results were ob-

tained when the number of precision digits was set equal to $2N$. For our error estimates, we use Duffy's measure, the L and L_e defined as

$$L = \sqrt{\sum_{i=1}^{30} \frac{[f(t_i) - \tilde{f}(t_i)]^2}{30}} \quad (3.28)$$

and

$$L_e = \sqrt{\frac{\sum_{i=1}^{30} [f(t_i) - \tilde{f}(t_i)]^2 e^{-t_i}}{\sum_{i=1}^{30} e^{-t_i}}} \quad (3.29)$$

where $f(t)$ is the analytical solution and $\tilde{f}(t)$ is the numerical solution. Hence L is the root-mean-square error and L_e is the same as L but weighted by the factor e^{-t} [40].

All computations were performed using a 64-bit operating system with an Intel(R) Core(TM) i7-8550u CPU processor. The algorithms were implemented in Maple 2018 using Maple's digits command to set the required precision.

Standard double precision

Tables 4.4 and 4.5 show that when compared with the other four algorithms, the Fourier series method performs with the least accuracy on all the functions tested. It also fails to reconstruct functions 8, 15, 17 and 18, with poor results for functions 4, 5 and 12.

However, for the functions which it successfully reconstructs, it does so with an RMS accuracy of between $L = 3.6(-5)$ and $1.2(-2)$. We believe that this scheme will improve greatly when an acceleration scheme is applied. This is an issue we intend to investigate in future work.

With the exception of the function 7, $J_0(t)$, Logan's algorithm successfully inverts all the functions given in Tables 4.2 and 4.3 with very good accuracy. We

found that in SDP best results are obtained by equating $a = 1$ in (3.17). Tables 3 and 4 show that for these functions the RMS error varies between $3.6(-8)$ to $8.4(-12)$.

Except for function 7 $J_0(t)$, the Abate-Valko scheme successfully inverts all the functions in Tables 4.3 and 4.4. Moreover, it does so with greater accuracy than the Logan scheme. The tables show that the RMS error varied between $6.5(-11)$ and $6.2(-12)$.

Tables 4.3 and 4.4 show that the Stehfest algorithm shows poor accuracy when inverting functions 1, 7, 10 and 11. For these functions the RMS error varies between $2.0(-2)$ to $9.2(-3)$. Its poor performance is due to the fact that the Stehfest algorithm has difficulty inverting functions of a cyclic nature [99]. However, it inverts the remaining functions with good accuracy with an RMS error of between $2.9(-5)$ to $0.0(0)$. Tables 4.3 and 4.4 shows that the Salzer-Gaver algorithm shows poor accuracy for functions 1, 7, 10 and 11. These are the very same functions that the Stehfest algorithm has problems inverting. Again this is due to the difficulties it encounters when inverting cyclic functions. It inverts the remaining functions with less accuracy than the Stehfest, with an RMS error varying between $10(-15)$ to $10(-5)$.

Multi precision

With the exception of function 7, the Logan and Abate-Valko algorithms successfully inverted the remaining functions to a high degree of accuracy. Duffy [40] also remarks that when using the Talbot contour, he had difficulties accurately inverting the Bessel function. This may be related to the combination of the singularity on the imaginary axis and the branching nature of the square root function.

Abate-Valko[1] state that they were able to overcome this by increasing the weights and hence the precision as a function of t . However, we were unable to replicate their results for this function.

Overall, the Abate-Valko scheme showed far greater accuracy than Logan's across all the functions tested. However, Logan's algorithm was still able to produce highly accurate results with RMS errors varying between $10(-60)$ to $10(-63)$. Moreover, Table 4.8 shows that Logan's scheme was able to perform the inversion of these functions with shorter elapsed times.

The Stehfset and Salzer-Gaver algorithms were able to invert all the functions to a high degree of accuracy. The Salzer-Gaver scheme was, in general, about twice as accurate as the Stehfset algorithm, which was less accurate than Abate-Valko's scheme. Nevertheless, the Stehfset scheme inverted the functions well within any generally desired accuracy, with the RMS error varying from $10(-41)$ to $0.0(0)$. Moreover, as Table 4.8 shows in terms of the elapsed time, it was the fastest of all the algorithms, for the most part twice as fast as the Abate-Valko scheme, which in turn was at least twice as fast as the Salzer-Gaver scheme.

Function No.	$f(s)$	$f(t)$
1	$\frac{1}{1+s^2}$	$\sin(t)$
2	$\frac{1}{(s+1)^2}$	te^{-t}
3	$\frac{1}{s^2}$	t
4	$\frac{1}{\sqrt{s}}$	$\frac{1}{\sqrt{(\pi t)}}$
5	$\frac{\ln s}{s}$	$-(\ln t + \gamma)$
6	$\frac{1}{s}$	1
7	$\frac{1}{\sqrt{s^2+1}}$	$J_0(t)$
8	$\frac{e^s K(1,s)}{s}$	$\sqrt{t(t+2)}$
9	$\frac{1}{s+0.5}$	$e^{-\frac{t}{2}}$
10	$\frac{1}{(s+0.2)^2+1}$	$e^{-0.2t} \sin(t)$
11	$\arctan(\frac{1}{s})$	$\frac{\sin(t)}{t}$

Table 3.2: Test Functions

Function No.	$f(s)$	$f(t)$
12	$\frac{1}{\sqrt{s}+\sqrt{s+1}}$	$\frac{1-e^{-t}}{2\sqrt{\pi t^3}}$
13	$\frac{1}{\sqrt{s}(1+\sqrt{s})}$	$e^t \operatorname{erfc}(\sqrt{t})$
14	$e^{-2\sqrt{s}}$	$\frac{e^{-\frac{1}{t}}}{\sqrt{\pi t^3}}$
15	$\frac{e^{-\frac{1}{4s}}}{s^{\frac{3}{2}}}$	$\frac{2 \sin(\sqrt{t})}{\pi}$
16	$\log(1 + \frac{1}{s})$	$\frac{1-e^{-t}}{t}$
17	$\frac{\arccos(s-1)}{\sqrt{s(s-2)}}$	$e^t K(0, t)$
18	$\frac{e^{-\frac{1}{s}}}{\sqrt{s}}$	$\frac{\cos(2\sqrt{t})}{\sqrt{(\pi t)}}$
19	$\frac{1}{\sqrt{s+\sqrt{s^2+1}}}$	$\frac{\sqrt{2} \sin(t)}{2t^{\frac{3}{2}} \pi}$

Table 3.3: Test Functions Continued

	Fourier		Logan		Valko		Stehfest		Salzer-Gaver	
Function	L	L_e	L	L_e	L	L_e	L	L_e	L	L_e
1	1.5(-4)	2.8(-4)	3.7(-9)	1.2(-11)	3.0(-11)	9.4(-14)	1.4(-3)	2.6(-5)	2.0(-2)	9.1(-4)
2	6.1(-4)	1.4(-4)	1.8(-9)	2.7(-10)	1.0(-11)	3.9(-14)	8.9(-6)	8.9(-6)	2.8(-6)	3.2(-6)
3	1.2(-3)	1.2(-3)	7.5(-9)	2.3(-11)	4.4(-11)	1.4(-13)	7.0(-8)	7.1(-8)	1.1(-8)	1.1(-8)
4	7.3(-2)	8.4(1.0)	7.4(-9)	2.3(-11)	4.8(-11)	1.7(-13)	5.4(-8)	6.2(-7)	2.8(-6)	3.2(-6)
5	6.8(-2)	6.9(-2)	4.1(-11)	1.2(-11)	6.4(-12)	2.2(-14)	2.2(-8)	1.4(-7)	2.3(-5)	8.5(-5)
6	6.1(-4)	6.1(-4)	7.5(-9)	2.3(-11)	6.5(-11)	2.1(-13)	0.0(0)	0.0(-)	8.7(-15)	4.8(-14)
7	2.8(-4)	3.4(-4)	Fail	Fail	Fail	Fail	1.9(-2)	6.7(-3)	1.7(-2)	5.1(-3)
8	Fail	Fail	1.2(-8)	3.8(-11)	8.1(-11)	2.7(-13)	9.2(-8)	1.2(-7)	6.3(-4)	6.3(-4)
9	5.7(-4)	4.7(-4)	4.6(-9)	1.6(-11)	3.3(-11)	1.2(-13)	1.2(-6)	4.7(-6)	1.9(-8)	2.8(-7)
10	6.1(-4)	1.9(-4)	3.1(-9)	9.5(-12)	3.0(-11)	9.7(-14)	9.2(-3)	4.8(-3)	5.2(-4)	2.9(-5)
11	2.8(-4)	3.8(-4)	5.9(-9)	1.8(-11)	3.4(-11)	1.2(-13)	7.4(-3)	2.7(-3)	6.5(-3)	1.9(-3)

Table 3.4: Standard Double Precision

	Fourier		Logan		Valko		Stehfest		Salzer-Gaver	
Function	L	L_e	L	L_e	L	L_e	L	L_e	L	L_e
12	2.0(-1)	4.2(-1)	3.1(-9)	9.5(-12)	1.6(-11)	6.2(-14)	6.2(-8)	2.0(-8)	8.2(-8)	1.4(-6)
13	3.1(-3)	1.6(-3)	3.7(-9)	1.2(-11)	2.7(-11)	8.9(-14)	8.6(-7)	4.0(-7)	3.3(-7)	3.7(-7)
14	6.2(-5)	3.6(-5)	3.6(-8)	1.2(-7)	6.2(-12)	1.3(-13)	1.3(-6)	1.7(-5)	1.6(-6)	2.2(-5)
15	Fail	Fail	3.3(-9)	1.0(-11)	3.9(-11)	1.2(-13)	5.7(-7)	1.0(-7)	7.9(-6)	1.9(-5)
16	5.8(-4)	4.3(-4)	5.2(-9)	1.6(-11)	2.5(-12)	9.1(-14)	2.2(-8)	1.4(-7)	4.9(-7)	8.8(-6)
17	Fail	Fail	1.2(-8)	3.9(-11)	7.3(-11)	2.5(-13)	6.0(-8)	3.0(-6)	2.6(-6)	1.1(-5)
18	Fail	Fail	2.7(-9)	8.4(-12)	1.7(-11)	6.3(-14)	2.9(-5)	1.6(-6)	5.9(-6)	3.4(-6)

Table 3.5: Standard Double Precision Continued.

	Logan		Valko		Stehfest		Salzer-Gaver	
Function	L	L_e	L	L_e	L	L_e	L	L_e
1	6.2(-63)	1.1(-63)	6.9(-110)	1.3(-110)	6.1(-41)	8.5(-43)	7.6(-124)	1.0(-125)
2	7.5(-63)	1.2(-63)	6.0(-110)	1.2(-110)	7.1(-77)	9.7(-79)	1.4(-184)	2.0(-184)
3	7.5(-63)	1.2(-63)	6.9(-110)	1.3(-110)	5.0(-92)	5.0(-92)	3.9(-184)	3.9(-184)
4	1.2(-60)	2.5(-60)	3.3(-107)	6.9(-107)	6.0(-94)	7.0(-93)	5.2(-132)	6.0(-132))
5	1.1(-60)	1.7(-60)	2.0(-118)	2.8(-118)	4.8(-93)	2.6(-92)	1.1(-293)	2.0(-292)
6	3.6(-61)	3.6(-61)	4.5(-108)	4.5(-108)	0.0(0)	0.0(0)	2.0(-293)	1.1(-292)
8	8.3(-62)	3.4(-62)	3.6(-119)	1.5(-119)	2.0(-72)	7.4(-74)	3.8(-133)	1.1(-132)
9	3.6(-61)	3.6(-61)	4.5(-108)	4.5(-108)	2.5(-94)	3.5(-93)	6.0(-182)	2.1(-184)
10	6.5(-63)	1.1(-63)	6.8(-110)	1.2(-110)	1.1(-45)	1.5(-47)	1.1(-128)	1.5(-130)
11	3.6(-61)	3.6(-61)	4.5(-108)	4.5(-108)	1.2(-42)	1.6(-44)	3.1(-126)	4.2(-128)

Table 3.6: Multi-Precision $N = 200$.

	Logan		Valko		Stehfest		Salzer-Gaver	
Function	L	L_e	L	L_e	L	L_e	L	L_e
12	5.9(-61)	1.3(-60)	6.9(-110)	1.3(-110)	6.1(-41)	8.5(-43)	3.4(-132)	4.4(-131)
13	3.0(-61)	3.4(-61)	6.0(-110)	1.2(-110)	7.1(-77)	9.7(-79)	1.6(-133)	2.9(-133)
14	7.4(-63)	2.0(-64)	6.9(-110)	1.3(-110)	5.0(-92)	5.0(-92)	3.4(-102)	6.5(-101)
15	6.7(-62)	2.7(-62)	3.3(-107)	6.9(-107)	6.0(-94)	7.0(-93)	1.3(-133)	2.6(-133))
16	3.6(-61)	3.6(-61)	2.0(-118)	2.8(-118)	4.8(-93)	2.6(-92)	2.5(-131)	3.3(-130)
17	1.4(-60)	2.0(-60)	2.3(-107)	3.1(-107)	0.0(0)	0.0(0)	8.5(-131)	3.5(-132)
18	1.1(-60)	2.5(-60)	3.2(-107)	6.9(-107)	1.4(-41)	1.9(-43)	6.3(-132)	6.8(-131)
19	8.3(-62)	3.4(-62)	3.6(-119)	1.5(-119)	2.0(-72)	7.4(-74)	3.8(-133)	1.1(-132)

Table 3.7: Multi-Precision Continued $N = 200$.

	Logan		Valko		Stehfest		Salzer-Gaver	
Function	w	τ	w	τ	w	τ	w	τ
2	255	0.89	77	1.44	202	0.43	100	2.37
8	17(-5)	0.55	9(-7)	0.88	12(-5)	0.41	10	3.06
11	175	0.94	167	3.87	320	1.74	160	2.56
13	29	0.72	12(-5)	0.94	20	0.41	12	0.99
18	35	0.56	16	1.05	46	0.52	20	1.27

Table 3.8: Elapsed time τ in seconds

3.6 Summary

In standard-double-precision, the Abate-Valko algorithm provides the best results for the numerical reconstructions for the functions tested in this paper. The Fourier algorithm had the worst performance of the five algorithms tested. Both the Stehfest and Salzer-Gaver algorithms had difficulty reconstructing functions of a cyclic nature. None of the algorithms was able to invert the $J_0(t)$ function accurately.

In multi-precision, The Stehfest and the Salzer-Gaver schemes inverted all the functions with high accuracy. The Logan and Abate-Valko schemes were only able to invert the $J_0(t)$ with limited accuracy. However, they were both able to reconstruct all the other functions with a high degree of accuracy. The most accurate algorithm in multi-precision was the Salzer-Gaver scheme. However, as Table 4.8 shows, it also had the longest elapsed times. On the other hand, the Stehfest algorithm had the shortest elapsed times for the selected functions in Table 4.8. The algorithms that used the Abate-Valko were the most accurate, but Logan could reconstruct the functions with shorter elapsed times. Therefore we conclude that when working in standard precision, Valko's algorithm

performed best. However, in multi-precision, the Stehfest algorithm is best as it inverted all the functions with a high degree of accuracy and the shortest elapsed times.

The next chapter uses regularisation information to solve the Fisher KPP (Kolmogorov, Petrovsky, Piskunov) reaction-diffusion equation. This problem has inherent perturbation issues and will require an inversion algorithm best suited to handle the additional noise introduced by the perturbation. The information on noise and the best precision environments of the algorithms tested in this and chapter 2 will be used in solving this equation.

Chapter 4

A Laplace Transform Finite Difference Scheme for the Fisher-KPP Equation.

4.1 Introduction

In chapters 2 and 3, the behaviour of some of the main algorithms for numerically inverting the Laplace transform were investigated for their handling of noise and their performance in precision environments. In this chapter, the knowledge gained in those investigations will be used for a specific application of the LTFDM for solving an equation in which perturbations can adversely affect the accurate reconstruction of the exact solution.

This chapter presents a numerical approach to the solution of the Fisher-KPP (Kolmogorov, Petrovsky, Piskunov) reaction-diffusion equation in which the spatial variable is developed using a purely finite difference scheme and the time

development is obtained using a hybrid Laplace Transform Finite Difference Method (LTFDM). The travelling wave solutions usually associated with the Fisher-KPP equation are, in general, not deemed suitable for treatment using Fourier or Laplace transform numerical methods. However, we were able to obtain accurate results when some degree of time discretisation is inbuilt into the process. While this means that the advantage of using the Laplace transform to obtain solutions for any time t is not fully exploited, the method does allow for considerably larger time steps than is otherwise possible for finite-difference methods.

(This chapter was published as a research paper: First published March 28, 2021, Research Article <https://doi.org/10.1177>

4.2 Fisher's equation

Fisher [48] suggested the equation,

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2} + Ku(1 - u) \quad (4.1)$$

to describe the propagation of a favourable gene in an infinitely long domain. The equation models the diffusion of an advantageous gene in a 1D habitat. A very informative discussion of the Fisher equation as it relates to propagation is given by [8] [72]. The expression in equation (4.1) combines the logistic and diffusion equations to simulate the respective processes of population growth and random dispersal of the advantageous gene under consideration. Thus D and K in (4.1) are the usual positive parameters associated with the diffusion and logistic equations.

Since its original development, the Fisher-KPP equation has been used exten-

sively to describe a wide variety of processes, including biology, chemical kinetics, auto-catalytic chemical reactions, branching Brownian motion, flame propagation, neurophysiology, the evolution of a neutron population in a nuclear reactor and chemical wave propagation [107].

The solution $u(x, t)$ of (4.1) represents the *proportion* of the mutant gene at a point x in its domain at some time t . Hence we must have that,

$$0 \leq u(x, t) \leq 1 \quad (4.2)$$

Fisher showed that (4.1) together with the additional boundary conditions,

$$u(-\infty, t) = 1 \quad \text{and} \quad u(+\infty, t) = 0 \quad (4.3)$$

exhibit travelling wave solutions of the form,

$$u(x, t) = u(x - ct) \quad (4.4)$$

moving at constant speed c in the positive x direction provided

$$c \geq C_{min} \quad (4.5)$$

when $C_{min} = 2\sqrt{KD}$ [77].

Thus the Fisher-KPP equation has an infinite number of travelling wave solutions, each moving with a wave speed $c \geq 2$. The solutions of $u(x, t)$ then connects the steady-state solution at $u = 1$ and the steady-state solution at $u = 0$. In keeping with the analysis of these steady-state conditions, $u = 1$ is stable, while $u = 0$, unstable.

It is worth noting that analytical solutions of the Fisher-KPP equation exist for only a small class of problems and hence the importance of developing efficient

numerical schemes to obtain solutions to (4.1) .

Although Fisher proposed his model for the wave advancement of an advantageous gene in 1937, it was not until 1974 that numerical solutions to the equation began to appear. The first of which was the seminal paper by Canosa [17] who used the Accurate Space Derivative method (ASD), sometimes referred to as the pseudo-spectral approach. Since then, many researchers have investigated numerical solutions to equation (4.1) for which Anjal et al. give a comprehensive summary of the main contributions [107]. However, these methods all incorporate some small time discretisation process, which requires iterations of the algorithm at each time step. As we discuss in the next section, our proposed solution to (4.1) allows us to obtain accurate results with considerably larger discretisation in the time domain.

In developing a numerical approach to solve the Fisher-KPP equation, we needed to keep two important points in mind. First, Canosa [17] showed that all waves are stable against *small* local perturbations but linearly unstable against general perturbations of *infinite* extent. This sensitivity to perturbations of infinite extent is essential for us because, as we explain in ‘Numerical examples and discussion’, the LTFDM involves inversion procedures which can introduce perturbations into the solution.

The second point is that Canosa was able to demonstrate by a simple stability analysis that computation is unstable against round-off errors building up at the leading tail of the waves [17]. We were able to overcome this difficulty by a particular application of the inversion process for the LTFDM.

4.3 The Laplace Transform Finite Difference Method

We consider an approach to the numerical solution of the Fisher-KPP equation (4.1) in which the spatial variable is discretised using a purely finite difference approach, and the temporal variable is removed by using a hybrid LTFDM. The significant advantage of this method is that it eliminates the time dependency parameter and the associated discretisations which are necessary to obtain solutions at a particular time t .

When using finite difference and other time discretisation methods to solve differential equations, for implicit schemes, the size of the time step is limited by the stability conditions required for convergence of the finite difference scheme. In linear cases, this usually involves hundreds and sometimes thousands of time steps to arrive at the solution for some desired time. Iterations are then required at each time step which involves using a variety of matrix methods to solve the vast systems of linear equations generated by the scheme.

For non-linear cases, this is compounded by the fact that a further iterative process is usually required at each time. Since each of these iterations introduces a certain amount of round-off and truncation error, careful consideration must be given to their control and management when implementing these schemes.

The Laplace transform has the potential to do away with time discretisation, and its associated error management by transforming the time domain into the Laplace space, s , via the integral transform,

$$\int_0^{\infty} f(t)e^{-st} dt = F(s) \quad (4.6)$$

Then computations done in the Laplace space, s , can be inverted back into the time domain at any desired time t . Hence the LTFDM can lead to the

required solution with virtually one-time step. By employing this method, we can potentially obtain substantial increases in speed and accuracy over traditional finite difference and time discretisation methods. With the additional benefit of reducing by one the dimensions of the governing equation, simplifying the resulting finite difference scheme needed to discretise the remaining spatial variable.

4.3.1 Inverting the data

The recovery of the function $f(t)$ is via the inverse Laplace transform which is most commonly defined by the Bromwich integral formula

$$\mathcal{L}^{-1}\{F(s)\} = f(t) = \frac{1}{2\pi i} \int_{u-i\infty}^{u+i\infty} F(s) e^{st} ds \quad (4.7)$$

for some u , where u is a real number, [97]. The choice of s in (4.6) and so in (4.7) is not an arbitrary one. If we choose s so that it lies on the positive real axis, we are treating the solution of (4.6) as a positive real integral equation. The problem here is that the inverse problem is known to be ill-posed, meaning that small changes in the values of $F(s)$ can lead to large errors in the values for $f(t)$ [10].

Hence when Laplace transform methods are employed for finding numerical solutions to partial differential equations, we must take account of the fact that the corresponding inversion methods can be highly sensitive to the inevitable noisy data that arises in their computation. This is a consequence of both truncation and round-off error, a process which is exacerbated for non-linear schemes. Our method attempts to mitigate these factors by employing the Fixed Talbot inversion algorithm. In our earlier work [38], we have shown that this inversion scheme reduces the effects that noisy data can have in adversely

perturbing the finite difference scheme. In this sense it can produce better results than the widely used Stehfest inversion method.

4.4 Method

We first non-dimensionalise equation (4.1) by letting

$$u = U\bar{u} \quad x = L\bar{x} \quad \text{and} \quad t = T\bar{t} \quad (4.8)$$

Employing the chain rule,

$$\frac{\partial}{\partial x} = \frac{1}{L} \frac{\partial}{\partial \bar{x}} \quad \text{and} \quad \frac{\partial}{\partial t} = \frac{1}{T} \frac{\partial}{\partial \bar{t}} \quad (4.9)$$

By letting $U = a$, $T = \frac{1}{Ka}$ and $L = \sqrt{\frac{D}{Ka}}$, and dropping the overbars (4.1) becomes,

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + u(1 - u) \quad (4.10)$$

with boundary conditions

$$u(-\infty, t) = 1 \quad \text{and} \quad u(+\infty, t) = 0 \quad (4.11)$$

The Laplace transform of the time derivative in (4.10) is

$$\mathcal{L}\left\{\frac{\partial u}{\partial t}\right\} = s\bar{u}(x, s) - u(x, 0) \quad (4.12)$$

where

$$\bar{u}(x, s) = \mathcal{L}\{u(x, t)\} \quad (4.13)$$

And the Laplace transform of the spatial derivative in (4.10) is

$$\mathcal{L}\left\{\frac{\partial^2 u}{\partial x^2}\right\} = \frac{d^2}{dx^2}\bar{u}(x, s) \quad (4.14)$$

However, it is well known that the Laplace transform cannot be successfully performed on non-linear governing equations, and so some linearisation process is necessary before the LTFDM can be implemented [119]. To overcome this, we follow Zhu et al. [117] who successfully applied the Laplace Transform dual reciprocity method to diffusion equations of the form,

$$\nabla^2 u = \frac{\partial u}{\partial t} - \beta f(u) \quad (4.15)$$

where β is a given constant and $f(u)$ is a non-linear function. Zhu first decomposed the function $f(u)$ in equation (4.15) into $f(\tilde{u})u$ then in order to find the solution of the unknown function at particular time t_1 equation 4.15 is linearised as

$$\nabla^2 u = \frac{\partial u}{\partial t} - \beta f(\tilde{u})u \quad (4.16)$$

in which $\tilde{u}$ is the previously iterated solution at time t_1 . Thus for equation (4.1) we would have,

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + u - u\tilde{u} \quad (4.17)$$

Then the Laplace transform of (4.17) is,

$$\bar{u}(x, s) - u(x, 0) = \frac{d^2}{dx^2} \bar{u}(x, s) + \bar{u}(1 - \tilde{u}) \quad (4.18)$$

with transformed boundary conditions,

$$\bar{u}(-\infty, t) = \frac{1}{s} \text{ and } \bar{u}(+\infty, t) = 0 \quad (4.19)$$

Using a central-difference scheme on the spatial derivative, the finite difference

scheme for (4.18) is,

$$\bar{u}_{i-1} - \bar{u}_i(2 + \delta x^2 s_j + \delta x^2 - \delta x^2 \tilde{u}_i) + \bar{u}_{i+1} = -\delta x^2 u(0)_i \quad (4.20)$$

where δx is the size of the spatial step in the x -direction, s_j is the j th Laplace parameter and $u(0)_i = u(x_i, 0)$. Then (4.20) can be expressed as the tridiagonal system,

$$\begin{pmatrix} a_{1j} & 1 & & 0 \\ 1 & a_{2j} & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & a_{n-1j} \end{pmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \dots \\ \dots \\ u_{n-1} \end{bmatrix} = - \begin{bmatrix} \delta x^2 u(x_1, 0) + \frac{1}{s_j} \\ \delta x^2 u(x_2, 0) \\ \dots \\ \dots \\ \delta x^2 u((x_{n-1}, 0)\Re) \end{bmatrix} \quad (4.21)$$

where

$$a_{ij} = 2 + \delta x^2 s_j + \delta x^2 - \delta x^2 \tilde{u}_i \quad (4.22)$$

in (4.21). After solving this tridiagonal system the data is then inverted to transition from the Laplace space, s , back into the time domain.

The Stehfest algorithm for numerically inverting the Laplace Transform

In their wide-ranging study of algorithms for inverting the Laplace transform, Davies and Martin [36] cite the Stehfest algorithm [99] as providing accurate results on a wide variety of test functions. Since then, this algorithm has become widely used for inverting the Laplace Transform and is favoured due to its reported accuracy and ease of implementation.

The algorithm takes the transformed data in the Laplace space $F(s)$ and pro-

duces $f(t_1)$ for a specific time value $t = t_1$. Choosing

$$s_j = j \frac{\ln 2}{t_1}, \quad j = 1, 2, \dots, M, \text{ for } M \text{ even.} \quad (4.23)$$

The numerical inversion is given by

$$f(t) \approx \frac{\ln 2}{t} \sum_{j=1}^M A_j F(s_j) \quad (4.24)$$

with

$$A_j = (-1)^{\frac{M}{2}+j} = \sum_{k=\lfloor \frac{j+1}{2} \rfloor}^{\min(j, \frac{M}{2})} \frac{k^{\frac{M}{2}} (2k)!}{(\frac{M}{2} - k)! k! (k-1)! (j-k)! (2k-j)!} \quad (4.25)$$

Theoretically, $f(t)$ becomes more accurate for larger M , but the reality is that rounding errors worsen the results if M becomes too large. According to Stehfest, “The optimum M is approximately proportional to the number of digits the machine is working with” [99].

Also in our earlier work we found that the Stehfest algorithm does not handle noisy data well [38]. As we show in Section 5, this can have the effect of introducing perturbations into the travelling wave solutions of the Fisher-KPP equation.

The Fixed Talbot algorithm for numerical inversion the Laplace Transform

Here we use the function,

$$S(z) = \frac{z}{1 - e^{-z}} \quad (4.26)$$

which maps the closed interval $M = [-2\pi i, 2\pi i]$ on the imaginary z -plane onto the curve L in the s -plane giving the integral,

$$f(t) = \frac{1}{2\pi i} \int_L F(s) e^{st} ds \quad (4.27)$$

(See Logan [66] for the details of this transformation).

Next we follow the procedure as adopted by Logan for numerically integrating (4.27).

With $s = S(z)$ this becomes

$$f(t) = \frac{1}{2\pi i} \int_M [F(S(z)) e^{S(z)t} S'(z)] dz \quad (4.28)$$

where

$$S'(z) = \frac{1 - (1+z)e^{-z}}{(1 - e^{-z})^2} \quad (4.29)$$

and $M = [-2\pi, 2\pi]$. For convenience we write,

$$f(t) = \frac{1}{2\pi i} \int_M Q(z) dz \quad (4.30)$$

where

$$Q(z) = [F(S(z)) e^{S(z)t} S'(z)] \quad (4.31)$$

Then if we let $w = -iz$ for the integral in (4.30) so the interval of integration is now real and becomes $[-2\pi, 2\pi]$. Then using the trapezoid rule with n we obtain

$$f(t) \approx \frac{1}{n} \left\{ I(2\pi i) + I(-2\pi i) + 2 \sum_{j=1}^{n-1} I(iw_j) \right\} \quad (4.32)$$

where

$$w_j = 2\pi \left\{ \frac{2j}{n} - 1 \right\} \quad (4.33)$$

and we note that $I(2\pi i) = I(-2\pi i) = 0$ [66].

4.5 Numerical examples and discussion

Example 1

For our first example we use (4.10)

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + u(1 - u)$$

and its associated boundary conditions,

$$u(-\infty, t) = 1 \quad \text{and} \quad u(+\infty, t) = 0$$

Ablowitx and Zeppetella [2] give an exact solution for a particular wave speed

$c = \pm \frac{5}{\sqrt{6}}$ as

$$u(x, t) = \frac{1}{\left[1 + \exp \left(\sqrt{\frac{1}{6}} x - \left(\frac{5}{6} \right) t \right) \right]^2} \quad (4.34)$$

which we use to compare our numerical results with.

When we first implemented the LTFDM it produced distortions in the upper tail of the travelling wave for larger values of t . This is shown in Figure 3.1.

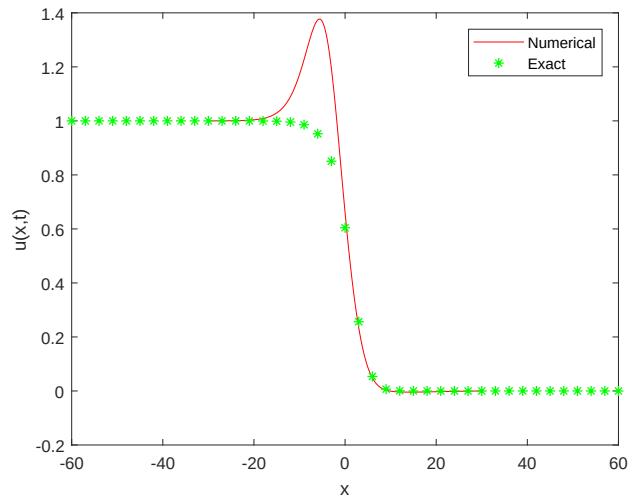


Figure 4.1: Profile without time discretisation. $t = 1.5$

We eventually surmised that these distortions were due to the existence of perturbations of infinite extent. In other words, the approximation of the initial condition on a finite domain. A stability analysis carried out by Gazdang et al. [51] showed that super speed waves or waves with speed greater than C_{min} could be maintained if subject only to infinitesimally small positive perturbations.

As is well known, the numerical inversion of the Laplace transform is a perturbed problem. Thus perturbations generated by the numerical scheme itself can then introduce noise into the inversion algorithms, which cannot be completely filtered out. However, we found that these perturbations can be reduced if some time discretisation, together with a reinitialisation of the initial condition, is introduced into the numerical method. While the full benefit of using the Laplace transform, i.e., to solve for any time t is partially diminished, introducing some measure of time discretisation meant we were able to use larger time steps than would be the case for other finite difference methods [93].

As we have shown in our previous work [38], the Fixed Talbot inversion method

is more efficient at filtering out this noise than the more widely used Stehfest algorithm. This is shown in Figures 3.2 and 3.3 where oscillations in the right-hand tail are present when using the Stehfest inversion method at time $t = 0.8$ but are absent in the Talbot inversion method.

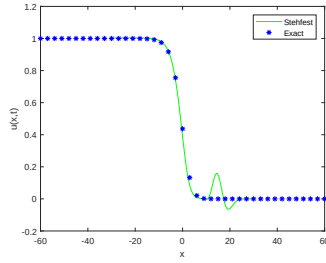


Figure 4.2: Profile using Stehfest. $t = 0.8$

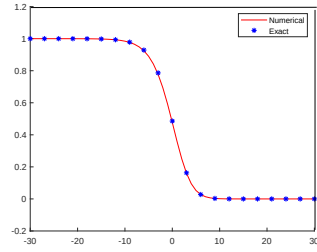


Figure 4.3: Profile using Talbot. $t = 0.8$

Thus smaller time steps are required for comparable accuracy for the Stehfest inversion than for the Talbot. Because of its inability to deal adequately with noisy data, the Stehfest algorithm is also sensitive to the spatial step size δx as smaller spatial discretisations can also introduce round-off error into the computations. Hence the choice of using the Talbot algorithm for carrying out the LTFDM inversion procedure. Details of this method on the field of study can be found in [38].

The Talbot algorithm is also very effective in dealing with the build-up of round-off error in the right tail of the waves. As Canosa points out, “This does not seem due to the numerical method used but to the physical nature of the problem described by the equation, which gives rise to an exponential growth of the solutions when this is exponentially small. This basic difficulty makes it difficult to do a rigorous simulation of the solutions of Fisher’s equation”.

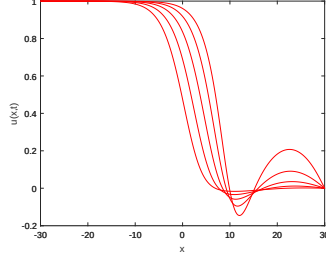


Figure 4.4: Talbot: $n = 55$ for $t = 1$ to 5.

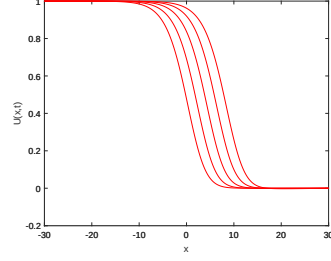


Figure 4.5: Talbot: $n = 555$ for $t = 1$ to 5

This effect is shown in Figure 3.4. However, we overcome this problem by merely increasing n (where n is the number of strips used in the trapezium method) in (4.32) from $n = 55$ to 555, which completely removes the instability and restores the travelling wave profile Figure 3.5. The restoration of the travelling profile is due to the Talbot algorithm's ability to filter out noise with increasing n [38]. While no exact solutions exist for (4.1) for wave speeds other than $c = \pm \frac{5}{\sqrt{6}}$ we can use the approximate asymptotic solution,

$$U(z) = (1 + e^{\frac{z}{c}})^{-1} + \frac{1}{c^2} e^{\frac{z}{c}} (1 + e^{\frac{z}{c}})^{-2} \log \left[\frac{4e^{\frac{z}{c}}}{(1 + e^{\frac{z}{c}})^2} \right] + O\left(\frac{1}{c^4}\right) \quad (4.35)$$

With $c \geq C_{min} = 2$ and $z = x - ct$, to test our numerical scheme for a variety of wave speeds. The accuracy of the asymptotic solution increases for large c [76].

Example 2.

Cattani, Carlo et al. [19] give an exact solution for the Fisher type equation,

$$\frac{\partial u}{\partial t} = v \frac{\partial^2 u}{\partial x^2} - bu^2 + au \quad (4.36)$$

where $0 < t \leq \infty$, $-\infty < x < \infty$ with the boundary condition

$$u(-\infty, t) = 0.5, \quad u(\infty, t) = 0 \quad (4.37)$$

and initial condition,

$$u(x, 0) = -\frac{1}{4} \frac{a}{b} \left[\operatorname{sech}^2 \left(-\sqrt{\frac{a}{24c}} x \right) - 2 \tanh \left(-\sqrt{\frac{a}{24c}} x \right) - 2 \right] \quad (4.38)$$

The exact solution is,

$$u(x, t) = -\frac{1}{4} \frac{a}{b} \left[\operatorname{sech}^2 \left(-\sqrt{\frac{a}{24c}} x + \frac{5a}{12} t \right) - 2 \tanh \left(-\sqrt{\frac{a}{24c}} x + \frac{5a}{12} t \right) - 2 \right] \quad (4.39)$$

Since no exact solution exists for all wave speeds for (4.36) we derived a perturbation solution to test the numerical scheme at a variety of wave speeds. The perturbation solution for this case is given as,

$$U(z) = \frac{1}{2} (1 + e^{\frac{z/2}{c}})^{-1} + \frac{1}{c^2} e^{\frac{z}{c}} (1 + e^{\frac{z/2}{c}})^{-2} \log \left[\frac{\sqrt{2} e^{\frac{z/8}{c}}}{(1 + e^{\frac{z/2}{c}})^2} \right] + O\left(\frac{1}{c^4}\right) \quad (4.40)$$

4.6 Results

In our investigations, we found our algorithm performs with equal accuracy for spatial steps $0 < \delta x \leq 1$ and with the Laplace transform used within the time steps $\Delta t = 0.1, 0.2$ and 0.4 . This shows that it is stable across a wide variety of parameters. (For all computations $n = 555$).

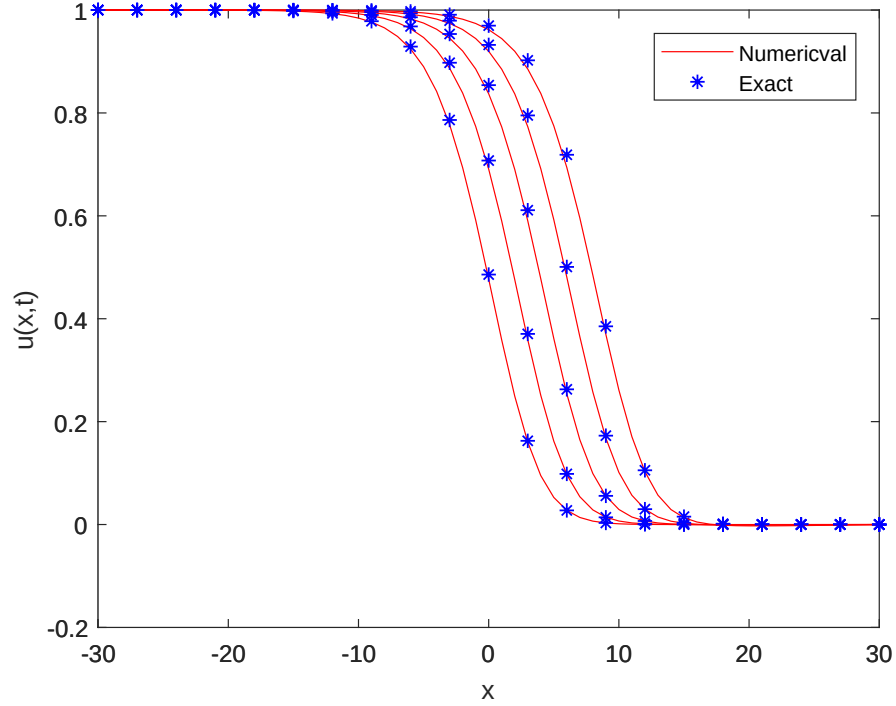


Figure 4.6: Profile $\Delta t = 0.2$, $t = 1$ to 5 .

Figure 3.6 shows the travelling wave profile for Example 1, compared with the exact solution [2]. The time discretisation used in the LTFDM is $\Delta t = 0.2$ with a spatial step of $\delta x = 0.1$. The numerical results show good agreement with the exact solution.

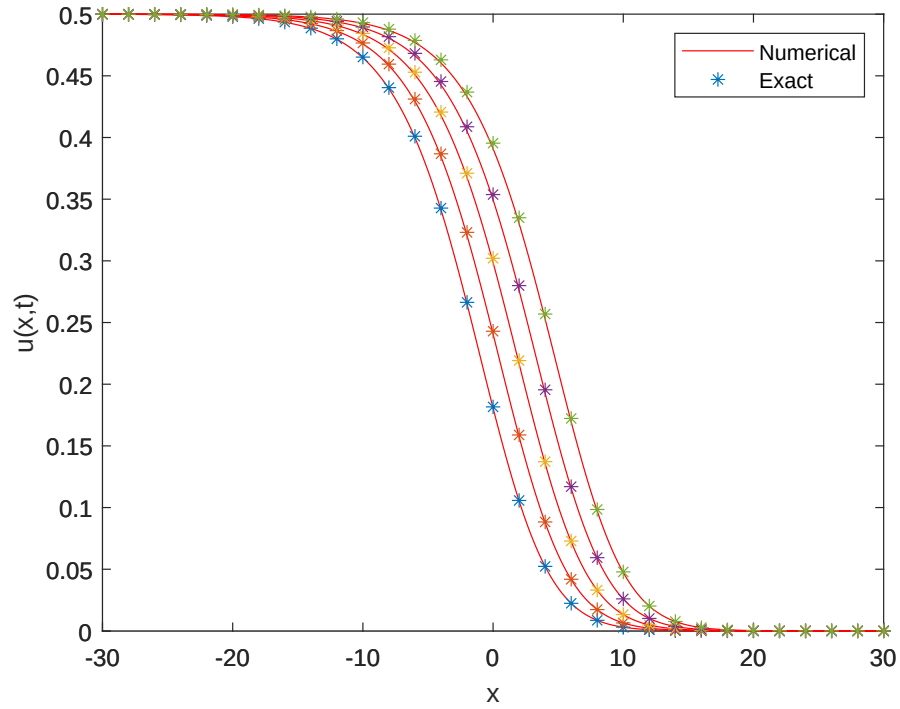


Figure 4.7: Profile $\Delta t = 0.2$, $t = 1$ to 5 .

Figure 3.7 shows the travelling wave profile for Example 2 with the exact solution [19]. The time discretisation used in the LTFDM is $\Delta t = 0.2$ with a spatial step of $\delta x = 0.1$. The numerical results show good agreement with the exact solution.

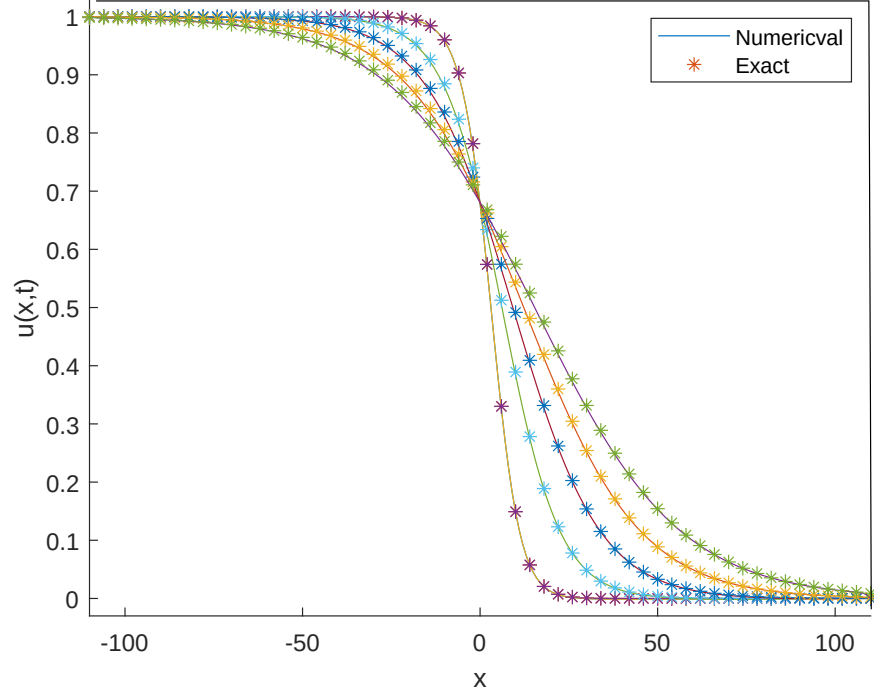


Figure 4.8: Profile Example 1, $\Delta t = 0.2$, $t = 1$, $C = 4 : 4 : 20$.

Figure 3.8 shows the travelling wave profile for example 1, compared with the perturbation solution for wave speeds $C = 4, 8, 12, 16, 20$. The time discretisation used in the LTFDM is $\Delta t = 0.2$ with a spatial step of $\delta x = 0.1$. The numerical results show good agreement with the exact solution.

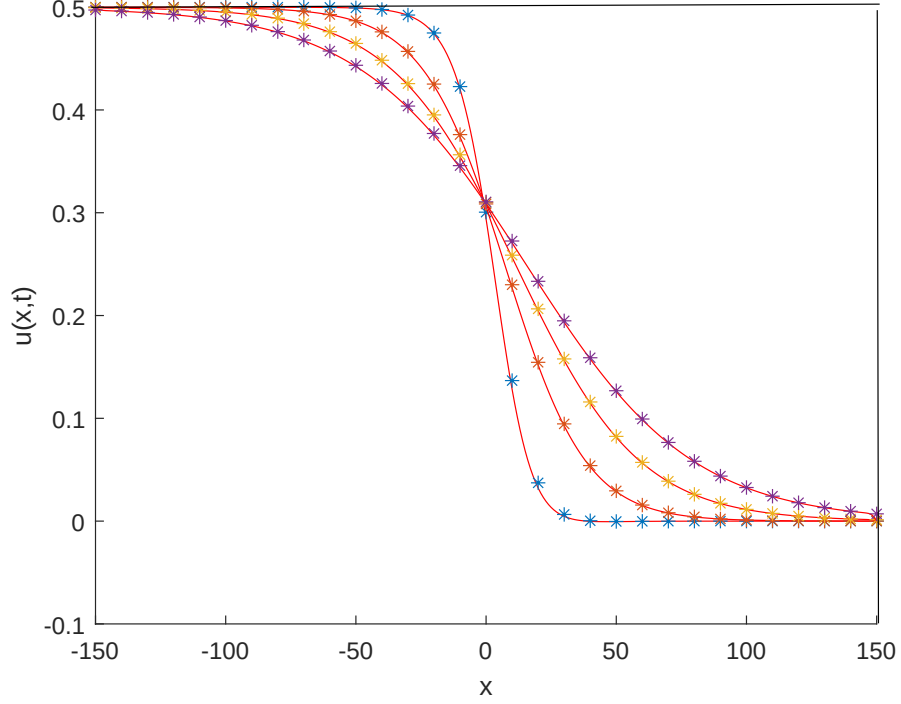


Figure 4.9: Profile Examole 2, $\Delta t = 0.2$, $t = 1$, $C = 4 : 4 : 20$.

Figure 3.9 shows the travelling wave profile for example 2 compared with perturbation solution (4.40) for wave speeds $C = 4, 8, 12, 16, 20$. The time discretisation used in the LTFDM is $\Delta t = 0.2$ with a spatial step of $\delta x = 0.1$. The numerical results show good agreement with the exact solution. Figures 3.10-3.12 show the error profile for example 1 for the wave speed shown.

Tables 3.1-3.3 present the results for Example 1 for times $t = 1$, $t = 2$, and $t = 4$. For all cases shown we set $\Delta t = 0.1$, $n = 555$, $\delta x = 0.25$, and $L = 60$.

For all the tables below the error is the absolute error.

x	Numerical	Exact	Error
-20	0.999742	0.999753	1.0 (−5)
-12	0.993285	0.995552	2.6 (−4)
-4	0.845643	0.849618	1.2 (−3)
2	0.252043	0.254227	8.6 (−3)
4	0.0956577	0.096161	5.2 (−3)
8	0.006466	0.006515	7.5 (−3)
12	0.000178	0.000284	3.7 (−4)

Table 4.1: Example 1, $t = 1$.

x	Numerical	Exact	Error
-20	0.999884	0.999893	8.8 (−6)
-12	0.996963	997190	2.3 (−4)
-8	0.984656	0.995740	1.1 (−3)
0	0.698929	0.707501	1.2 (−2)
4	0.255059	0.490844	1.5 (−2)
8	0.027968	0.028250	2.0 (−2)
10	0.006447	0.006719	2.1 (−2)

Table 4.2: Example 1, $t = 2$.

x	Numerical	Exact	Error
-20	0.999976	9999808	3.4 (−6)
-12	0.999379	0.999468	8.9 (−5)
-6	0.992958	0.993864	9.2 (−4)
2	0.846588	0.856205	1.1 (−2)
4	0.88938265	0.90291741	1.5 (−2)
6	0.490436	0.500723	2.1 (−2)
10	0.100190	0.103045	2.8 (−2)

Table 4.3: Example 1, $t = 4$.

Tables 3.4-3.6 present the results for problem 2 for times $t = 1$, $t = 2$, and $t = 4$. For all cases shown we set $\Delta t = 0.4$, $n = 555$, $\delta x = 0.25$, and $L = 60$.

x	Numerical	Exact	Error
-20	0.497713	0.497780	1.3 (−4)
-12	0.477704	0.478304	1.3 (−3)
-8	0.434286	0.435769	3.4 (−3)
-1	0.220496	0.98636	8.5 (−3)
4	0.0512810	0.051519	4.5 (−3)
15	0.000163	0.000162	6.2 (−3)
20	9.33(−6)	9.34(−6)	4.5 (−4)

Table 4.4: Example 2. $t = 1$.

x	Numerical	Exact	Error
-20	0.498548	0.497780	2.2 (−4)
-8	0.456965	0.459476	5.8 (−3)
-8	0.434286	0.435769	3.4 (−3)
01	0.238570	0.242958	1.8 (−2)
4	0.087159	0.088343	1.3 (−2)
15	0.000432	0.000432	7.9 (−2)
20	2.53(−5)	2.52(−5)	1.1 (−4)

Table 4.5: Example 2. $t = 2$.

x	Numerical	Exact	Error
-20	0.499320	0.499413	1.9 (−4)
-8	0.479300	0.481756	5.1 (−3)
-1	0.375701	0.383712	2.1 (−2)
01	0.238570	0.242958	1.8 (−2)
2	0.272176	0.279941	2.8 (−2)
15	0.002122	0.002123	2.3 (−2)
20	0.00013143	0.0001311	1.9 (−3)

Table 4.6: Example 2. $t = 4$.

For brevity we give a sample of the results in Tables 3.7-3.9 of the comparison of our method with the approximate perturbation solution for example 2 with $t = 1$. The length L is increased for higher wave speeds to ensure complete propagation of the wave as it moves to the right with increasing speed. In all cases $n = 555$, $\delta x = 1$, $\triangle t = 0.1$.

x	Numerical	Exact	Error
-124	0.499186.	0.499190	1.0 (−5)
-100	0.497323	0.497399	1.5 (−4)
-40	0.451696	0.451966	6.0 (−4)
-30	0.425057	0.425457	9.4 (−4)
10	0.214997	0.215853	4.0 (−3)
20	0.155856	0.156620	5.0 (−3)
50	0.009998	0.010083	8.5 (−3)

Table 4.7: Example 2 $t = 1$ $C = 10$.

x	Numerical	Exact	Error
-140	0.496314.	0.496336	4.4 (−5)
-100	0.486330	0.486413	1.6 (−4)
-40	0.414178	0.414573	6.0 (−4)
-20	0.425057	0.425457	9.5 (−4)
0	0.279545	0.280139	2.1 (−3)
40	0.1242715	0.124647	3.0 (−3)
80	0.039467	0.039600	8.5 (−3)

Table 4.8: Example 2 $t = 1$ $C = 15$.

x	Numerical	Exact	Error
-182	0.495876.	0.495901	4.9 (−5)
-100	0.469727	0.469891	3.5 (−4)
-10	0.310285	0.310802	1.7 (−3)
-20	0.425057	0.425457	9.5 (−4)
50	0.133185	0.133472	2.2 (−3)
70	0.090021	0.090213	2.1 (−3)
176	0.007480	0.007496	8.5 (−3)

Table 4.9: Example 2, $t = 1$ $C = 20$.

Figures 3.16-3.18 show the error profile comparing our scheme with the approximate perturbation solution at $t = 1$, for various wave speeds. The error profile and the corresponding range of errors remain unchanged for varying wave speeds, for example, 1, but the error decreases with increasing wave speed, for example 2.

The Figures 3.17-3.19 demonstrate the stability of the results for varying mesh size δx . For brevity we show this for problem 2 at wave speed $C = 20$ and for time $t = 1$ We were able to achieve the same error profile with these mesh sizes for all the problems investigated in this paper.

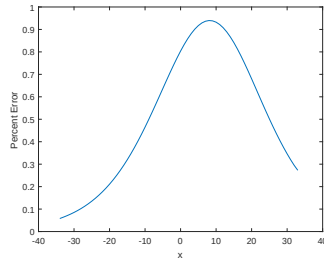


Figure 4.10: Example 1 Error Profile $\Delta t = 0.1$, $t = 1$, $C = 10$.

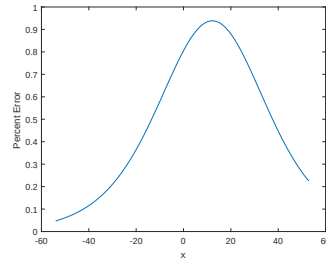


Figure 4.11: Example 1 Error Profile $\Delta t = 0.1$, $t = 1$, $C = 15$.

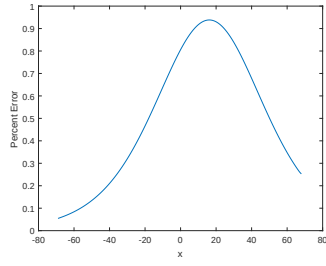


Figure 4.12: Example 1 Error Profile $\Delta t = 0.1$, $t = 1$, $C = 20$.

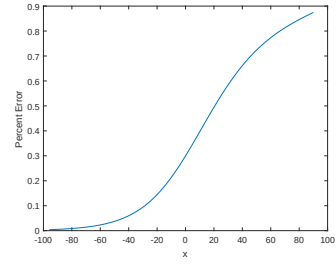


Figure 4.13: Example 2 Error Profile $\Delta t = 0.1$, $t = 1$, $C = 10$.

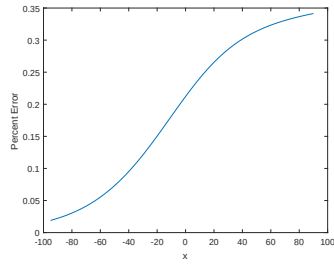


Figure 4.14: Example 2 Error Profile $\Delta t = 0.1$, $t = 1$, $C = 15$.

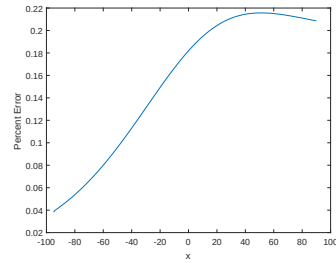


Figure 4.15: Example 2 Error Profile $\Delta t = 0.1$, $t = 1$, $C = 20$.

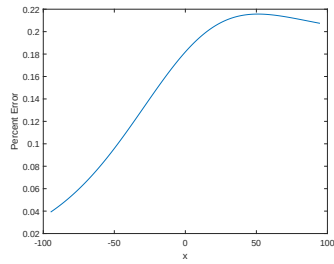


Figure 4.16: Example 2 Error Profile $\delta x = 0.5$.

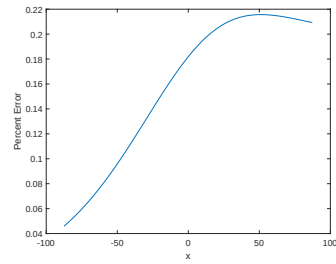


Figure 4.17: Example 2 Error Profile $\delta x = 0.25$.

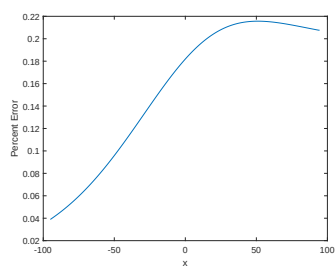


Figure 4.18: Example 1 Error Profile $\delta x = 0.05$.

4.7 Summary

This chapter proposes a numerical approach to the solution of the Fisher-KPP reaction-diffusion equation in which the spatial variable is discretised using a purely finite difference scheme, and the time development is obtained using a hybrid Laplace Transform Finite Difference Method (LTFDM). This method, to our knowledge, has not previously been applied to the Fisher-KPP equation, and Laplace transform methods are generally not deemed suitable for equations with travelling wave solutions.

However, by introducing some time discretisation into our LTFDM we were able to obtain results with less than one per cent error over a range of times, space and time discretisation, for various wave speeds. The time discretisation was necessary to reduce perturbations of infinite extent, which occur in numerical schemes for the Fisher-KPP equation. These perturbations can have a detrimental effect on the LTFDM since all the numerical schemes for inverting the Laplace transform are highly perturbed.

Thus crucial to the success of the method outlined in this paper is the choice of the Fixed Talbot inversion algorithm, which, as we have shown in our earlier work, is best at dealing with the inherent noise generated in finite difference schemes. This algorithm also had the effect of ironing out the build-up of round-off error in the right-hand tail of the travelling wave, a consequence of the physical nature of the problem.

The successful application of the LTFDM to a reaction-diffusion equation with travelling waves and inherent perturbation demonstrates the method's robustness. Thus in the next chapter, we extend the LTFDM to solve a series of linear and non-linear diffusion equations with a variety of initial and boundary conditions and test its performance against two popular finite difference methods used for solving diffusion equations.

Chapter 5

The Laplace Transform Finite Difference Method for Solving Linear and Non-Linear Diffusion Equations.

5.1 Introduction

This chapter uses the Laplace Transform Finite-Difference Method (LTFDM) to solve linear and nonlinear diffusion type problems with Dirichlet or first-type boundary conditions. The previous chapter employed the method to yield solutions to the fisher KPP reaction-diffusion equation. This chapter extends the method to other types of diffusion problems.

The main advantage of the LTFDM is the removal of time stepping procedures, which are usually associated with finite-difference methods. The LTFDM can

remove time step limits imposed on general finite difference schemes needed to control the accuracy and stability of the system. The solution at a particular time is not dependent on the result at any other time, apart from initial conditions, allowing a solution to be obtained in a one-time step.

This chapter uses the method to solve a series of one dimensional linear and nonlinear diffusion problems with various initial and boundary conditions. For each of the equations considered, the solution in the time domain is provided via the numerical inversion Laplace transform using the algorithms proposed by Talbot and Stehfest. The accuracy of these algorithms is then compared with Forward Time Central Space and Crank Nicholson Finite Difference schemes.

5.1.1 The Laplace transform

The Laplace transform is an integral transform defined as follows:

Let $f(t)$ be defined for $t \geq 0$, then the Laplace transform of $f(t)$ is given by,

$$\mathcal{L}\{f(t)\} = \int_0^{\infty} f(t)e^{-st} dt \quad (5.1)$$

Thus $\mathcal{L}\{f(t)\}$ is a function of s denoted as $F(s)$. The Laplace transform can be shown to exist for any function which can be integrated over any finite interval $0 < t < l$ for $l > 0$, and for which $f(t)$ is of exponential order, i.e.

$$|f(t)| < Me^{at} \quad (5.2)$$

as $t \rightarrow \infty$, where $M > 0$ is a finite real number and a is a small real positive number.

Analytically the inverse Laplace transform is usually obtained using the techniques of complex contour integration with the resulting set of standard transforms presented in tables [79].

The recovery of the function $f(t)$ is via the inverse Laplace transform, which is most commonly defined via the Bromwich contour integral.

$$L^{-1}\{F(s)\} = f(t) = \frac{1}{2\pi i} \int_{\alpha-i\infty}^{\alpha+i\infty} f(s) e^{st} ds \quad (5.3)$$

However, using the Laplace transform can generate data in the Laplace domain, which is not easily invertible to the real domain by analytical means. Thus numerical inversion techniques have to be used to convert the data from the s domain to the time domain [7], [41].

5.1.2 Laplace Transform Finite Difference Method

We consider an approach to the numerical solution for time-dependent diffusion-heat equations in which the spatial variable is developed using a purely finite difference approach. The time development is obtained using a hybrid Laplace Transform Finite Difference Method LTFDM. The significant advantage of this method is that it eliminates the time dependency parameter and the associated discretisation, which is necessary to obtain solutions at a particular time.

When using finite difference and other time discretisation methods to solve differential equations, the time step size is limited by the stability conditions required for convergence of the scheme [5], [102]. In linear cases, this usually involves hundreds and sometimes thousands of time steps to solve for some desired time. Iterations are then required at each time step which involves using various matrix methods to solve the vast systems of linear equations generated by the scheme.

For nonlinear cases, this is compounded by the fact that a further iterative process is usually required at each time step [74]. Since each of these iterations introduces a certain amount of round-off and truncation error, careful consideration must be given to their control and management when implementing these schemes.

The Laplace transform has the potential to do away with time discretisation and the associated error management by transforming the time domain into the Laplace space, s , via the integral transform in (5.1). The computations done in the Laplace space, s , can then be inverted back to the time domain for any desired time. Hence the LTFDM can lead to the required solution with virtually one-time-step. This means that stability and convergence problems caused by time discretisation are avoided, and the amount of computation is decreased without losing any essential characteristics of the problem [74]. Thus by em-

ploying the LTFDM, we can obtain substantial increases in speed and accuracy over traditional finite difference and time discretisation methods.

The LTFDM was first proposed by Moridis et al. [74] for the simulation of compressible liquid flow, where the Stehfest [99] algorithm was used to invert the data. Kang and Kwon [4] apply the LTFDM to the solution of the Black-Scholes equation and inverted the data from the Laplace space into the time domain by using the Fourier series method with an accelerated convergence scheme [28],[55], [42],[55]. Jacobs [57] used the LTFDM in the context of time-fractional heat equations with both Dirichlet and Neumann boundary conditions coupled with a compact fourth-order finite-difference scheme. Jacob's inversion scheme is one by [110], which is a particular application of the Bromwich contour integration method. However, this approach involves using three free parameters instead of the version used in this paper, namely the Fixed Talbot Algorithm, which uses a single parameter to invert the functions.

In 1994 Zhu et al. [119] combined the Laplace transform and the Dual Reciprocity Method (DRM) into what they called The Laplace Transform Dual Reciprocity Method (LTDRM) to solve linear time-dependent diffusion equations. In 1996 Zhu et al. [119] extended this method to solve a particular class of nonlinear diffusion equations. In 2005 Crann et al. [27] successfully applied this method to electromagnetic heating problems. In 2007 Davies and Crann [33] used a hybrid Laplace transform finite-difference boundary element method for solving diffusion equations. Zhu, Davies and Crann used the Stehfest inversion scheme to convert the data into the time domain.

Dimple Rani et al. used a numerical inverse Laplace transform for Solving a class of fractional differential equations [90]. The inversion method used an algorithm proposed by the authors based on Bernstein operational matrix [89]. Then in 2020, Dimple Rani et al. used the numerical inverse Laplace transform

based on Bernoulli polynomials operational matrix for solving nonlinear ordinary differential equations [88].

This paper applies the LTFDM to three linear and four nonlinear one-dimensional diffusion problems. The linear problems are chosen due to the variety of initial and boundary conditions. We then apply the LTFDM to solve four nonlinear diffusion problems. These problems are chosen because they have very different nonlinear terms. In problem four, we apply the LTFDM to the problem tackled by Zhu et al., who, as we stated before, used the LTDRM to solve this problem and compare our results. This problem has a non-linearity of u^3 . In problem five, the LTFDM is used to solve an equation whose nonlinear term is $\sqrt{1+u^2}$ and has hyperbolic boundary conditions. In problem six, the method is used on a diffusion equation with a u^2 nonlinear term. Finally, we apply the scheme to an equation with an exponential e^{-u} nonlinear term for problem seven. For all the nonlinear terms, we adopt a simple direct or semi-direct iterative linearisation procedure.

The authors are not aware of comparative studies for the LTFDM or other Laplace transform methods with finite-difference methods for one-dimensional diffusion equations in the literature. Two inversion schemes for both the linear and nonlinear problems are employed, the Fixed Talbot Algorithm [104] and the algorithm developed by Stehfest [99]. Using these two inversion schemes allows us to test the robustness of these algorithms, and we believe it is good practice for real-life applications of LTFDM where we do not have an exact solution to test the accuracy of the scheme.

In problems 1-3, the LTDFM methods are compared with the Forward Time Central Space (FTCS) finite difference scheme. However, this method fails to produce accurate results for the nonlinear problems 7 to 9 where we employ the more stable Crank-Nicholson finite-difference method [69], [114].

5.2 The Inversion Algorithms

5.2.1 Stehfest

In their wide-ranging study of algorithms for inverting the Laplace transform, Davies and Martin [36] cite the Stehfest algorithm [99] as providing accurate results on a wide variety of test functions. Since then, this algorithm has become widely used for inverting the Laplace Transform, being favoured due to its reported accuracy and ease of implementation.

The algorithm takes the transformed data in the Laplace space $F(s)$ and produces $f(t_1)$ for a specific time value $t = t_1$. Choosing

$$s_j = j \frac{\ln 2}{t_1}, \quad j = 1, 2, \dots, M, \text{ for } M \text{ even.} \quad (5.4)$$

The numerical inversion is given by

$$f(t) \approx \frac{\ln 2}{t} \sum_{j=1}^M A_j F(s_j) \quad (5.5)$$

with

$$A_j = (-1)^{\frac{M}{2}+j} = \sum_{k=\lfloor \frac{j+1}{2} \rfloor}^{\min(j, \frac{M}{2})} \frac{k^{\frac{M}{2}} (2k)!}{(\frac{M}{2} - k)! k! (k-1)! (j-k)! (2k-j)!} \quad (5.6)$$

Theoretically, $f(t)$ becomes more accurate for larger M , but the reality is that rounding errors worsen the results if M becomes too large. According to Stehfest, “The optimum M is approximately proportional to the number of digits the machine is working with” [99].

The Fixed Talbot algorithm for numerically inverting the Laplace Transform

Here we use the function,

$$S(z) = \frac{z}{1 - e^{-z}} \quad (5.7)$$

which maps the closed interval $M = [-2\pi i, 2\pi i]$ on the imaginary z -plane onto the curve L in the s -plane giving the integral,

$$f(t) = \frac{1}{2\pi i} \int_L F(s) e^{st} ds \quad (5.8)$$

(See Logan [66] for the details of this transformation).

Next we follow the procedure as adopted by Logan for numerically integrating (5.8).

With $s = S(z)$ this becomes

$$f(t) = \frac{1}{2\pi i} \int_M [F(S(z)) e^{S(z)t} S'(z)] dz \quad (5.9)$$

where

$$S'(z) = \frac{1 - (1+z)e^{-z}}{(1 - e^{-z})^2} \quad (5.10)$$

For convenience we write,

$$f(t) = \frac{1}{2\pi i} \int_M Q(z) dz \quad (5.11)$$

where

$$Q(z) = F(S(z)) e^{S(z)t} S'(z) \quad (5.12)$$

and $M = [-2\pi, 2\pi]$. Then if we let $w = -iz$ for the integral in (5.11) so the interval of integration is now real and becomes $[-2\pi, 2\pi]$. Then using the trapezoid rule with n we obtain

$$f(t) \approx \frac{1}{n} \left\{ (I(2\pi i) + I(-2\pi i) + 2 \sum_{j=1}^{n-1} I(iw_j)) \right\} \quad (5.13)$$

where

$$w_j = 2\pi \left\{ \frac{2j}{n} - 1 \right\} \quad (5.14)$$

and we note that $I(2\pi i) = I(-2\pi i) = 0$ [66].

5.3 Diffusion Equations.

We seek solutions to the general diffusion equation [96]

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2} + F(u) \quad (5.15)$$

where κ is the diffusivity constant and $F(u)$ can be equal to zero, a linear or a non-linear functions of u . For our first set of linear diffusion problems we have that $F(u) = 0$. in (5.15) and constant boundary conditions, $u(0, t) = c_1$ and $u(L, t) = c_2$. So we have,

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2} \quad (5.16)$$

This models for example the diffusion of heat along a one-dimensional bar of length L with initial condition $u(x, 0) = f(x)$ for $0 \leq x \leq L$ and constant boundary conditions for $t \geq 0$. The Laplace transform of the time derivative in (5.16) is

$$\mathcal{L} \left\{ \frac{\partial u}{\partial t} \right\} = s\bar{u}(x, s) - u(x, 0) \quad (5.17)$$

where

$$\bar{u}(x, s) = \mathcal{L}\{u(x, t)\} \quad (5.18)$$

And the Laplace transform of the spatial derivative in (5.16)

$$\mathcal{L}\left\{\frac{\partial^2 u}{\partial x^2}\right\} = \frac{d^2}{dx^2} \bar{u}(x, s) \quad (5.19)$$

With $\kappa = 1$, our finite difference scheme for (5.16) in the Laplace s -space is

$$\bar{u}(x, s) - u(x, 0) = \frac{d^2}{dx^2} \bar{u}(x, s) \quad (5.20)$$

with transformed boundary conditions,

$$\bar{u}(0, s) = \frac{c_1}{s_j} \quad \text{and} \quad \bar{u}(L, s) = \frac{c_2}{s_j} \quad (5.21)$$

Where c_1 and c_2 are constants. Then using a central-difference scheme on the spatial derivative, the finite difference scheme for (5.20) is,

$$\bar{u}_{i-1} - \bar{u}_i(2 + \delta x^2 s_j) + \bar{u}_{i+1} = -\delta x^2 u(0)_i \quad (5.22)$$

where δx is the step size in the x -direction, s_j is the j th Laplace parameter and $u_{0_i} = u(x_i, 0)$. To avoid the absolute error ‘blowing up’ for values of u near zero unless otherwise stated, we give the maximum relative error calculated along the entire length of the bar. In problems 1 to 3, we compare the results for the LTFDM with the Forward Time Central Space (FTCS) finite difference method [96]. Stability criteria requires that $r = \frac{\delta t}{\delta x^2} < \frac{1}{2}$. We found that a value of $r = 0.2$ was more than adequate to generate accurate results.

As far as the authors are aware, comparisons of the LTFDM with the FTCS and Crank-Nicholson finite-difference, at least for one-dimensional diffusion equations, have not been done. These results, therefore, should indicate the viability of using the LTFDM in these circumstances, as opposed to general finite-difference methods.

For all the examples studied in this paper we used $L = 1$, $\delta x = 0.1$ and $M = 12$ weights for the Stehfest inversion and $n = 555$ for the Talbot inversion. For brevity and ease of presentation in our tables, we use the notation $2.7(-3)$ instead of 2.7×10^{-3} . For all the examples, the error norm is the absolute percentage error given by:

$$E_{max} = \max \left| \frac{f_{numerical}(t_i) - f_{exact}(t_i)}{f_{exact}(t_i)} \times 100 \right|, \quad i = 1, \dots, 40 \quad (5.23)$$

(All computations were carried out using Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz)

5.3.1 Problem 1

This is equation (5.16) with $\kappa = 1$, initial condition $u(x, 0) = 0$ for $0 \leq x \leq L$ and boundary conditions $u(0, t) = u(L, t) = 1$, for $t \geq 0$. The exact solution is

$$u(x, t) = 1 + \sum_{n=1}^{\infty} \frac{2n}{\pi} (-1)^n e^{(-n^2 \pi^2 t)} \sin(n\pi x) \quad (5.24)$$

Table 5.1 shows that the FTCS method performs better for the times given than both LTDFM schemes, with the Talbot giving better results than the Stehfest inversion. This is also shown in Figure 1, which shows the temperature profile at $t = 1$.

5.3.2 Problem 2

This is equation (5.16) with $\kappa = 0.5$, $u(x, 0) = T_0$, $u(0, t) = T_1$, $u(x, t) \rightarrow T_0$ as $x \rightarrow \infty$. The exact solution is given by

$$T_0 + (T_1 - T_0) \operatorname{erf} \left(\frac{x}{2 \sqrt{\kappa t}} \right) \quad (5.25)$$

Here we use $T_0 = 2$ and $T_1 = 4$.

t	FTCS	Stehfest	Talbot
0.5	3.1(-5)	1.2(-4)	4.2(-5)
1	3.1(-7)	6.2(-5)	3.0(-6)
2	7.0(-11)	3.6(-5)	4.6(-10)
3	6.5(-15)	2.1(-5)	1.2(-13)
4	9.1(-18)	1.9(-5)	2.6(-13)
5	4.7(-22)	1.1(-5)	6.2(-13)

Table 5.1: Problem 1 Error

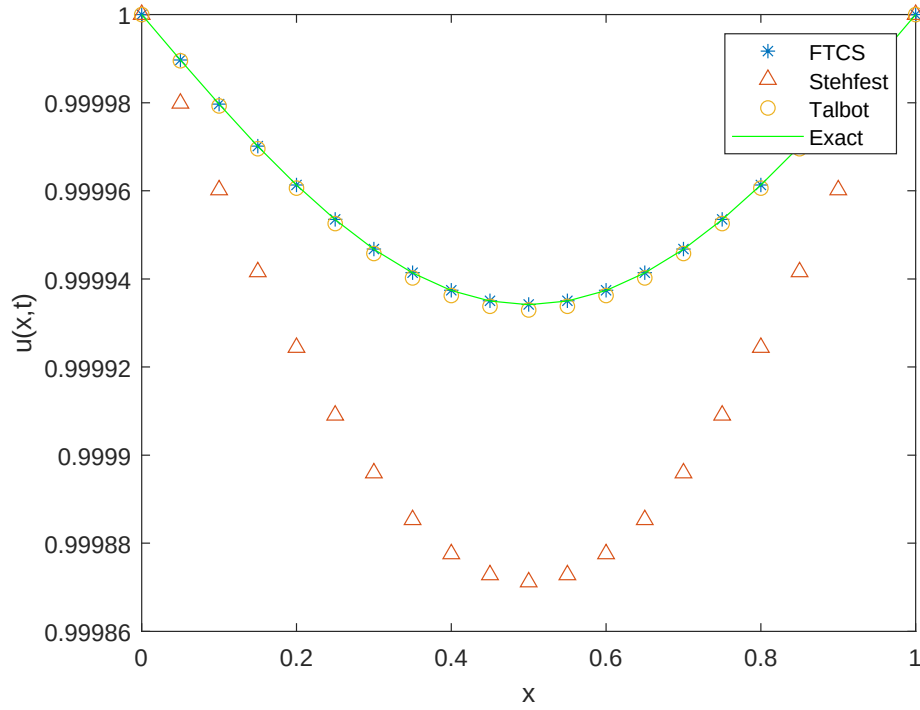


Figure 5.1: Temperature Profile Problem 1 for $t = 1$.

t	FTCS	Stehfest	Talbot
0.5	2.0(-3)	1.9(-2)	1.1(-3)
1	9.8(-4)	9.7(-3)	5.4(-4)
2	4.9(-4)	4.8(-3)	2.7(-4)
3	3.3(-4)	3.2(-3)	1.8(-4)
4	2.4(-4)	2.4(-3)	1.6(-4)
5	2.0(-4)	1.9(-3)	1.1(-4)

Table 5.2: Problem 2 Error

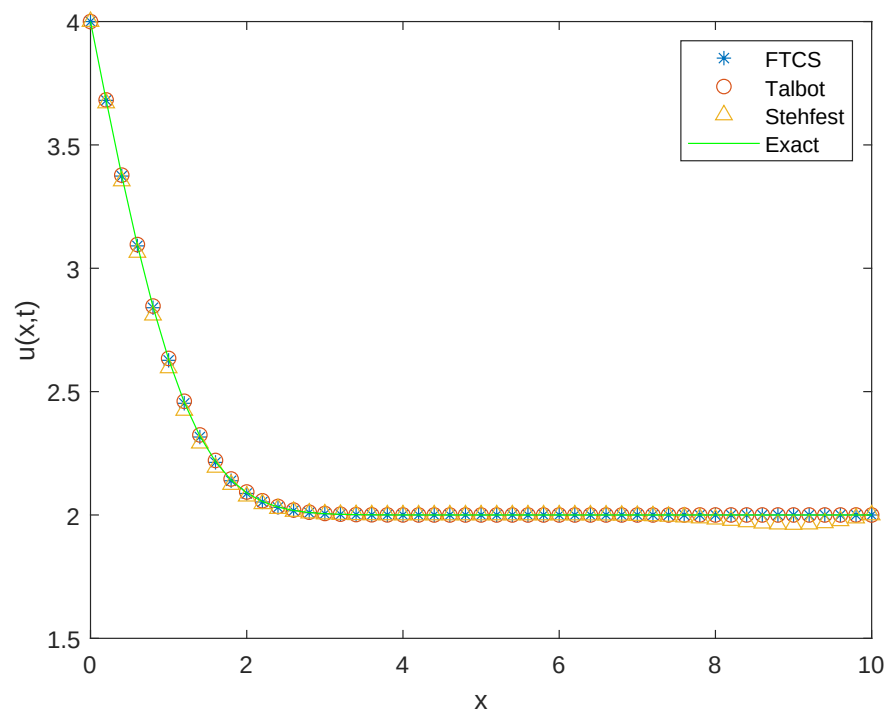


Figure 5.2: Temperature Profile Problem 2 for $t = 1$.

Table 5.2 shows that the LTFDM with Talbot inversion gives the most accurate results for times considered performing slightly better than the FTCS and the LTDFM Stehfest inversion schemes. For all the methods, the accuracy increases at larger times. Figure 5.2 shows the temperature profile at $t = 1$, with the Stehfest LTFDM increasing in error for x between 8 and 10.

5.3.3 Problem 3

This equation is the linear cyclic diffusion equation

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2} - \frac{\pi}{\alpha} x \cos(\pi t) \quad (5.26)$$

with initial condition $u(x, 0) = \sin(\pi x)$, the left hand boundary at $u(0, t) = 0$, and a cyclic boundary condition at $u(1, t) = \sin(\pi t)$ for $t > 0$. The exact solution is given by

$$u(x, t) = \exp(-\alpha \pi^2 t) \sin(\pi x) + x \sin(\pi t) \quad (5.27)$$

The finite-difference scheme in the Laplace space is

$$\bar{u}_{i-1} + \bar{u}_i \left(-2 - \frac{\delta x^2}{2} s_j \right) + \bar{u}_{i+1} = \delta x^2 \left(u_{0_i} - \frac{\pi}{\alpha} x \frac{s_j}{s_j^2 - \pi^2} \right) \quad (5.28)$$

where δx is the size of the spatial step in the x -direction, s_j is the j th Laplace parameter and $u_{0_i} = u(x_i, 0)$.

It is well known that the Stehfest algorithm has difficulty reconstructing the sine and cosine functions for any $t \geq \frac{\pi}{2}$, [62], [65]. This is shown in Figure 5.3 and Figure 5.4.

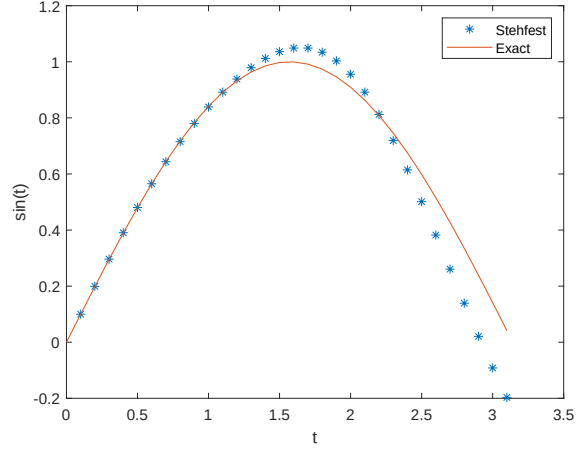


Figure 5.3: Stehfest reconstruction of $\sin(t)$.

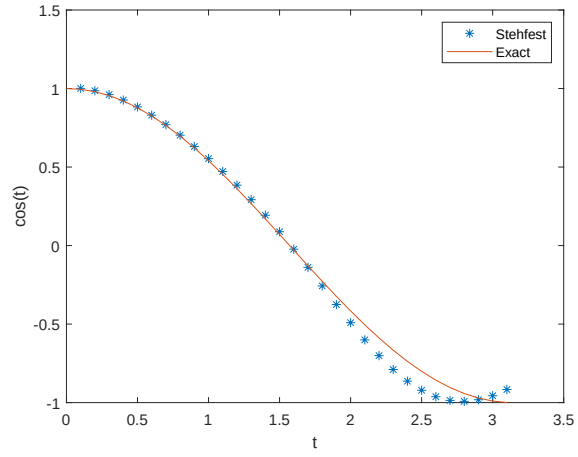


Figure 5.4: Stehfest reconstruction of $\cos(t)$.

This can be overcome by implementing the inversion scheme in a multi-precision environment [18], [113] or by a piecewise reconstruction of the function [27]. However, as our emphasis in this paper is on simple comparative implementation for solving diffusion type equations, we choose instead to evaluate the boundary condition and the cosine function on the right-hand side of (5.26) at the desired

time and then invert these numerical results as constants in the Laplace space.

$$\sin(\pi t) = \frac{a_t}{s_j} \text{ and } \cos(\pi x t) = \frac{b_t}{s_j} \quad (5.29)$$

(For all the tables presented the first or t column indicates the interval used in the evaluation. Thus $0, 2\pi$ means $\forall t \in [0, 2\pi]$).

t	FTCS	τ	Stehfest	τ	Talbot	τ
$0, 2\pi$	8.7(-5)	3.6(1)	9.6(-1)	.4.0(-2)	.3.1(-1)	1.7(-1)
$1, 3\pi$	8.7(-5)	8.1(0)	1.4(-1)	5.1(-2)	3.5(-2)	2.3(-1)
$2, 4\pi$	7.3(-5)	1.4(1)	1.6(-1)	5.3(-2)	6.1(-3)	2.4(-1)
$3, 5\pi$	6.4(-5)	2.2(1)	3.1(-3)	5.5(-2)	6.8(-4)	2.6(-1)
$4, 6\pi$	6.3(-5)	3.2(1)	3.1(-2)	5.7(-2)	1.2(-4)	3.6(-1)
$5, 6\pi$	6.3(-5)	4.1(1)	3.1(-2)	5.2(-1)	1.3(-5)	3.3(-1)

Table 5.3: Problem 3 Error $\alpha = 0.2$

t	FTCS	Stehfest	Talbot
$0, 2\pi$	4.1(-5)	7.3(0)	2.0(0)
$1, 3\pi$	4.1(-5)	8.4(-2)	1.0(-3)
$2, 4\pi$	4.1(-5)	1.2(-3)	1.0(-5)
$3, 5\pi$	4.1(-5)	1.2(-3)	5.0(-8)
$4, 6\pi$	4.1(-5)	1.2(-3)	5.0(-8)
$5, 7\pi$	4.1(-5)	1.2(-3)	5.0(-8)
$6, 8\pi$	1.4(-5)	1.2(-3)	4.7(-8)

Table 5.4: Problem 3 Error $\alpha = 0.5$

t	FTCS	Stehfest	Talbot
$0,2\pi$	2.7(-5)	6.1(-2)	2.1(-1)
$1,3\pi$	2.3(-5)	6.6(-4)	1.0(-5)
$2,4\pi$	2.3(-5)	6.0(-4)	8.1(-10)
$3,5\pi$	2.3(-5)	6.0(-4)	8.1(-10)
$4,6\pi$	2.3(-5)	6.0(-4)	1.2(-8)
$5,7\pi$	2.3(-5)	6.0(-4)	1.2(-7)

Table 5.5: Problem 3 Error $\alpha = 1$

Figures 5.5, 5. 6, and 5.7 show the temperature profiles for problem 3 with $\alpha = 1$ and $t \in [0, 4\pi]$,

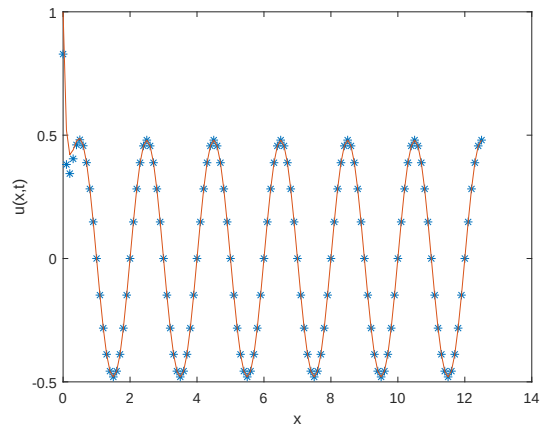


Figure 5.5: Temperature Profile Problem 3 using Talbot inversion

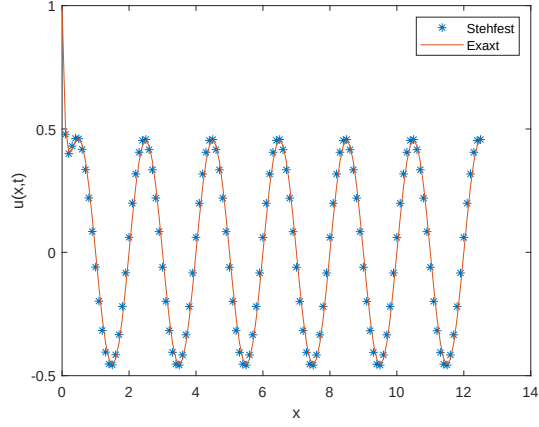


Figure 5.6: Temperature Profile Problem 3. Stehfest inversion

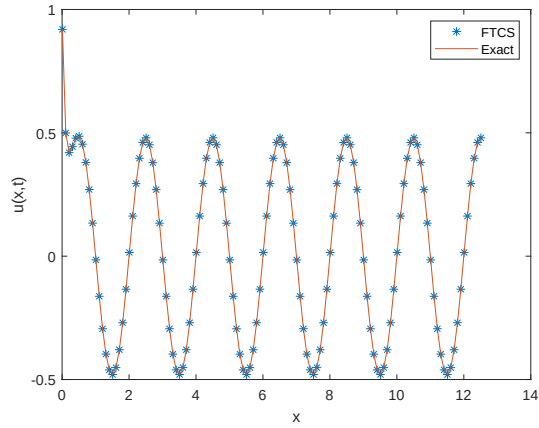


Figure 5.7: Temperature Profile Problem 3. FTCS

The results from Tables 5.3, 5.4, and 5.5 show the FTCS performing better at earlier times. At these times, the Talbot inversion has difficulty dealing with the jump discontinuity shown in Fig 5.7, but this is also the case for the Stehfest algorithm for values of α other than 1. However, the Talbot scheme steadily improves with time, doing slightly better than the FTCS in the interval $t \in [5\pi, 7\pi]$. Tables 5.4 and 5.5 show that for the larger α values, the FTCS

maintains the same relative error for all times while the Talbot steadily improves, doing considerably better than the FTCS and the Stehfest for greater times. Table 5.3 also shows that the FTCS scheme had elapsed times τ , of several orders larger than the LTFDM. This verifies the points made earlier in section 1.2 about the LTFDM ability to save computational time in arriving at the solution offering a considerable advantage in using the LTFDM over time discretisation schemes.

5.4 Non-Linear Diffusion Type problems

5.4.1 Problem 4

Our first non-linear problem is taken from Zhu et al. [119] who used the Laplace Transform Dual Reciprocity method to solve

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t} + u(1-u)(u-\gamma) \quad (5.30)$$

However, it is well known that the Laplace transform cannot be successfully performed on non-linear governing equations, and so some linearisation process is necessary before the LTFDM can be implemented, [47]. We use the same linearisation procedure adopted by Zhu. In order to find the solution of the unknown function at a particular time t_1 , Zhu linearised (5.30) as

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t} + u(\gamma - (\gamma + 1)\tilde{u} + \tilde{u}^2) \quad (5.31)$$

where $\tilde{u}$ is the previously iterated solution. The governing equation has the nonlinear terms u^2 and u^3 . Zhu linearises the u^2 term as $u\tilde{u}$ and the u^3 term

as $u\tilde{u}^2$. In the Laplace space the corresponding finite difference scheme is

$$\bar{u}_{i-1}^m - \bar{u}_i^m \left(2 + \delta x^2 s_j + \delta x^2 - \frac{\delta x^2}{2} \right) + \bar{u}_{i+1}^m = -\delta x^2 \left(u_{0_i} + \frac{1}{s_j} \right) \quad (5.32)$$

where δx is the size of the spatial step in the x -direction, s_j is the j th Laplace parameter and $u_{0_i} = u(x_i, 0)$. The initial condition is $u = 0$ and the boundary conditions are chosen to satisfy the exact solution

$$u = \frac{e^{\eta_1} + \gamma e^{\eta_2}}{1 + \eta_1 + \gamma e^{\eta_2}} \quad (5.33)$$

with

$$\eta_1 = \frac{1}{\sqrt{2}} \left(x - \left[\sqrt{2}\gamma - \frac{1}{\sqrt{2}} \right] \right) \quad (5.34)$$

and

$$\eta_2 = \frac{\gamma}{\sqrt{2}} \left(x - \left[\sqrt{2}\gamma - \frac{\gamma}{\sqrt{2}} \right] \right) \quad (5.35)$$

Since no Laplace Transform exists for (5.33) at the boundaries we evaluate its numeric value a say for specific t and express the transform at the boundaries as a/s_j for $x = 0$ and for $x = L$.

t	FTCS	Stehfest	Talbot
0.1	1.1(-2)	2.0(0)	2.0(0)
0.5	2.0(-2)	3.4(0)	3.5(0)
1	4.0(-2)	1.3(0)	1.2(0)
3	5.0(-4)	5.1(-2)	9.8(-2)
4	1.2(-4)	1.2(-2)	1.1(-2)
5	2.7(-5)	2.7(-3)	1.4(-3)

Table 5.6: Problem 4 Zhu Percentage Error

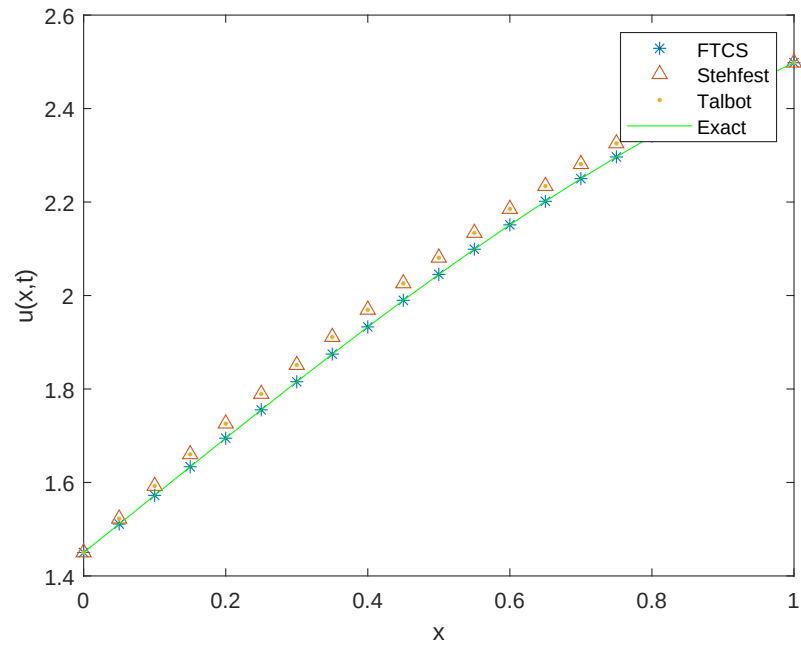


Figure 5.8: $t = 0.1$

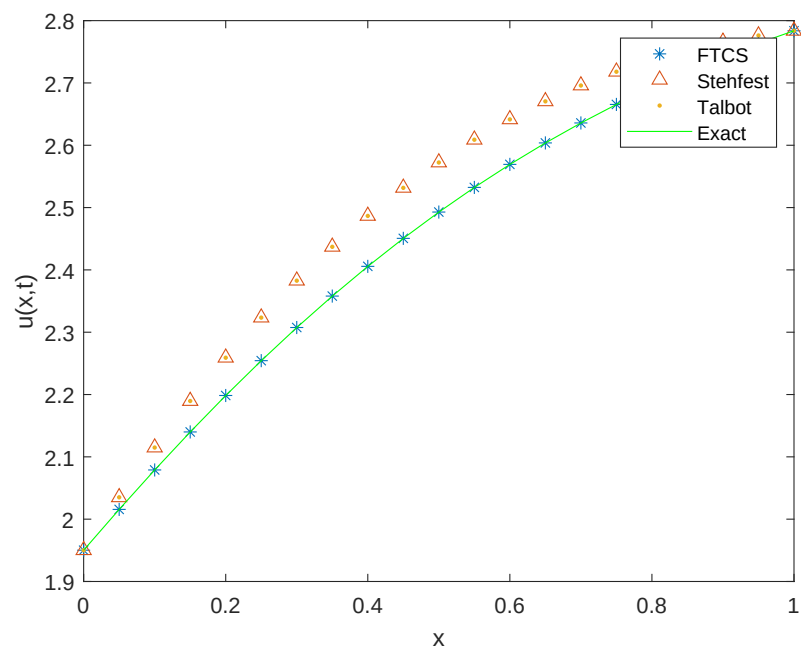


Figure 5.9: $t = 0.5$.

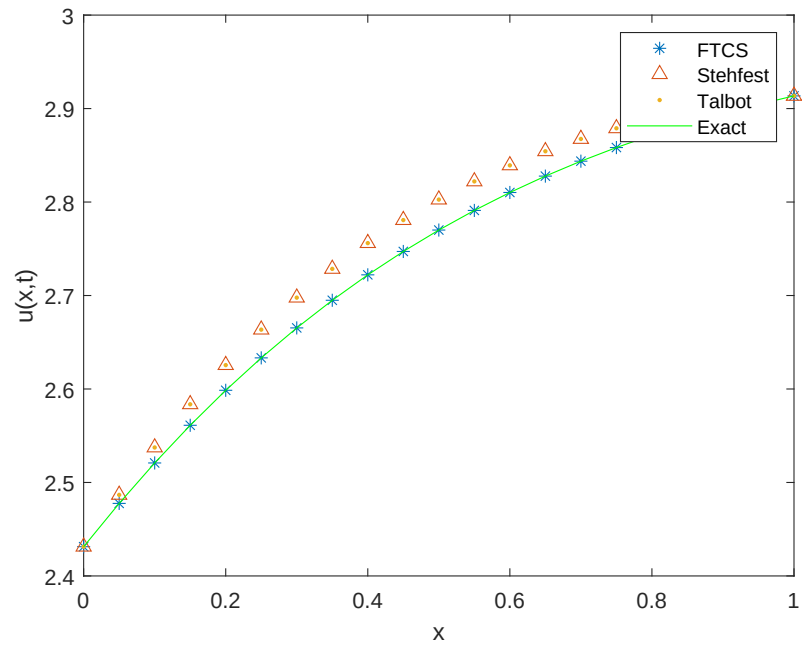


Figure 5.10: $t = 1.0$.

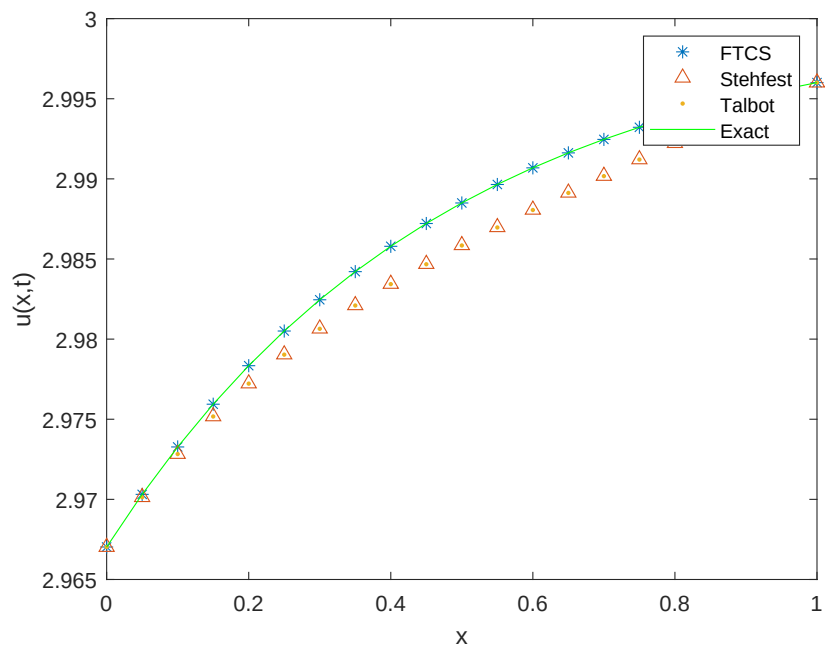


Figure 5.11: $t = 3.0$.

Table 5.6 shows that the FTCS performs better than both LTFDM schemes for all times shown. For this problem, we follow Zhu et al. and give the percentage error. The Talbot and Stehfest both give a percentage error greater than 3% at $t = 0.5$ compared with 0.0195% for the FTCS. All schemes then improve for greater t . However, the LTDFM performs better than the LTDRM as outlined by Zhu et al., who reports a maximum relative error at $t = 0.1$ of 3% and $t = 5$ compared with 1.9% for both LTFDM schemes and just less than 0.1% compared with 0.0014% for the Talbot inversion and 0.0027% for the Stehfest inversion.

Figures 5.8 to 5.11 show the temperature profile for problem 4 at the indicated times. All the graphs show the Talbot and the FTCS performing better than the Stehfest LTFDM.

5.4.2 Problem 5

Our next non-linear equation has the form

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t} + u - \sqrt{1 + u^2} \quad (5.36)$$

With initial and boundary conditions chosen to satisfy the exact solution $u(x, t) = \sinh(x + t)$.

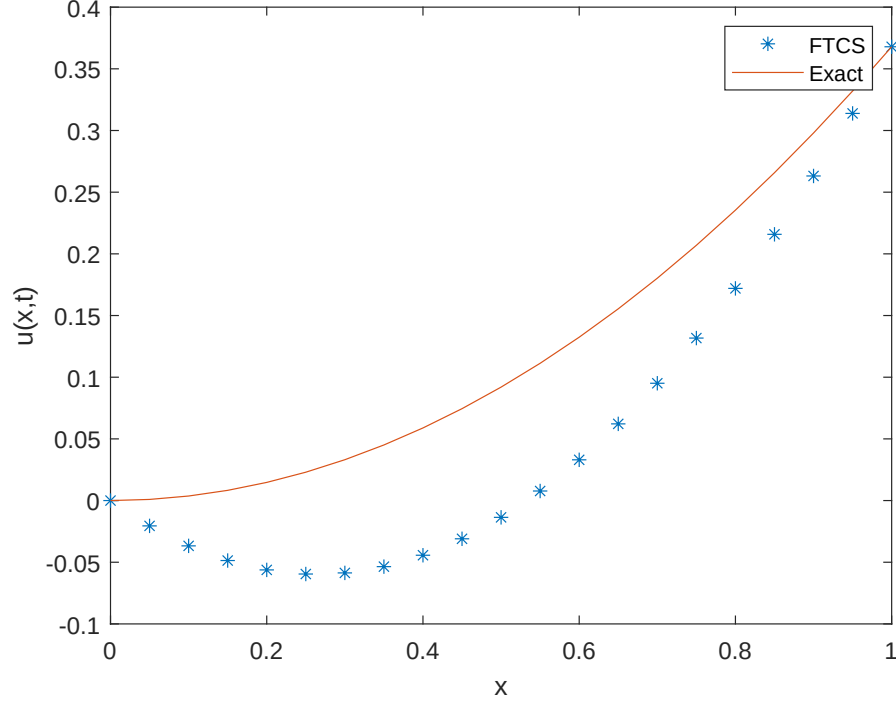


Figure 5.12: Temperature Profile Problem 5.

Figure 5.12 shows that the FTCS scheme fails to reconstruct the solution accurately. We therefore use the Crank-Nicolson method noted as a “unconditionally stable” finite-difference method [100], [96]. In this case, we use direct iteration to linearise $\sqrt{1+u^2}$ term. This means that the previously iterated solution is used for the u^2 term in equation(5.36). In the Laplace space, the corresponding finite difference scheme is

$$\bar{u}_{i-1}^m - \bar{u}_i^m \left(2 + \delta x^2 s_j + \delta x^2 \right) + \bar{u}_{i+1}^m = -\delta x^2 \left(u(0)_i + \sqrt{1 + (u_i^{m-1})^2} \right) \quad (5.37)$$

The Crank Nicholson scheme is

$$ru_{i-1}^j + u_i^j(2 - 2r - dt) + ru_{i+1}^j + \sqrt{1 + (u_i^{j,m-1})^2} + 2\delta t$$

The subscripts i and the subscript j denote the number of x and t intervals, respectively, and m and $m-1$ represent the current and previous iteration steps.

t	CN	Stehfest	Talbot
0.5	1.0(-2)	1.5(-1)	1.2(-2)
1	1.1(-2)	2.8(-4)	1.3(-4)
2	2.6(-2)	4.8(-4)	5.8(-4)
3	8.6(-2)	2.0(-2)	1.6(-3)
4	2.5(-1)	5.1(-3)	4.3(-3)
5	7.0(-1)	1.3(-2)	1.2(-3)

Table 5.7: Problem 5 Error

t	CN	τ	Stehfest	τ	Talbot	τ
0.5	1.0(-4)	1.8(0)	1.5(-2)	2.8(-2)	1.2(-2)	1.5(1)
1	2.8(-4)	7.0(0)	5.1(-4)	2.8(-2)	7.5(-5)	1.4(-1)
2	2.7(-4)	7.0(0)	8.5(-5)	2.9(-2)	5.9(-6)	1.5(-1)
3	9.1(-4)	1.0(1)	4.4(-4)	2.79(-2)	1.6(-5)	1.5(-1)
4	2.6(-3)	1.4(1)	7.7(-4)	3.1(-2)	4.4(-5)	1.5(-1)
5	7.3(-3)	1.7(1)	1.1(-3)	2.8(-2)	1.2(-4)	1.7(-1)

Table 5.8: Problem 5 Error $\delta x = 0.01$

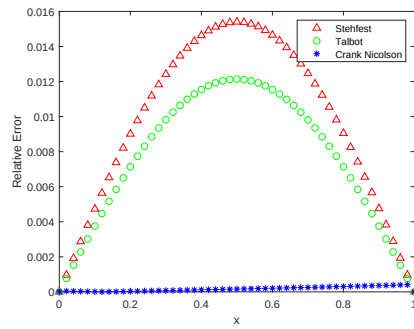


Figure 5.13: $t = 0.5$

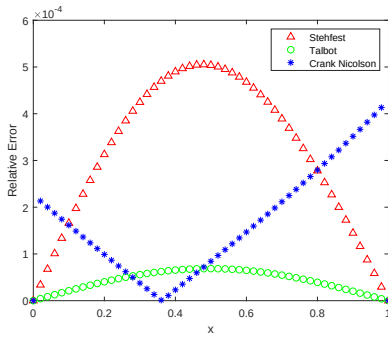


Figure 5.14: $t = 1.0$.

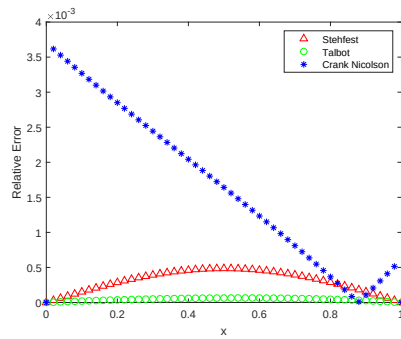


Figure 5.15: $t = 3.0$.

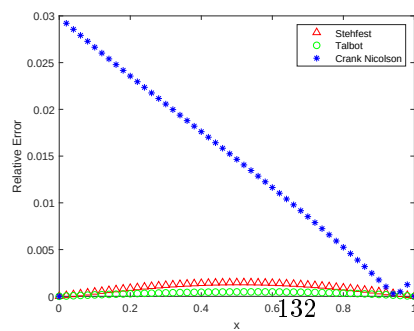


Figure 5.16: $t = 5.0$.

Problem 5.

Table 5.7 shows that Crank-Nicholson is not as accurate as the other schemes. Also, except for $t = 0.5$, the Talbot and the Stehfest LTFDM perform better than the Crank-Nicholson finite difference method. In Table 5.8, with a reduction in the spatial step from $\delta x = 0.1$ to $\delta x = 0.01$, we see an improvement in accuracy for all the schemes. However, there is a significant increase in the elapsed time τ for the Crank-Nicholson scheme. This substantiates the points we made in Section 1.2 about the advantages gained in speed and accuracy when employing LTFDM over traditional finite-difference methods. Figures 5.13 to 5.16 shows the error profiles for all three schemes with spacial step $\delta x = 0.05$ at times shown.

5.4.3 Problem 6

This problem has a nonlinearity of u^2 and is given by

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t} + u^2 + a(x, t) \quad (5.38)$$

with

$$a(x, t) = (2 + x^2)e^{-t} - x^4e^{-2t} \quad (5.39)$$

The boundary conditions are defined as follows:

$$u(0, t) = 0$$

$$u(1, t) = e^{-t}$$

with initial condition

$$u(x, 0) = x^2$$

For this case, we linearise the u^2 as $u^m u^{m-1}$ where m and $m-1$ represent the current and previous iteration steps, respectively. Our LTFDM scheme in the Laplace space is

$$\bar{u}_{i-1}^m - \bar{u}_i^m (2 + \delta x^2 s_j + \delta x^2 \bar{u}_i^{m-1}) + \bar{u}_{i+1}^m = \delta x^2 \left(-u(0)_i + \frac{2 + x^2}{s_j + 1} - \frac{x^4}{s_j + 2} \right) \quad (5.40)$$

and our Crank Nicholson finite difference scheme is

$$-ru_{i-1}^{j+1,m} + u_i^{j+1,m} (2 + 2r + \delta t u_1^{j+1,m-1}) - ru_{i+1}^{j+1,m} = ru_{i-1}^j + u_i^j (2 - 2r - \delta t u_i^{j,m-1}) + ru_{i+1}^j - 2\delta t a(x, t) \quad (5.41)$$

t	FTCS	CN	Stehfest	Talbot
0.5	1.6(-2)	5.2(-3)	1.7(-2)	1.2(-2)
1	1.1(-2)	2.0(-3)	1.0(-2)	3.0(-3)
2	4.4(-2)	1.1(-3)	3.9(-3)	4.1(-4)
3	1.6(-3)	4.3(-4)	1.5(-3)	5.6(-5)
4	6.1(-4)	1.5(-4)	5.2(-4)	7.1(-6)
5	2.3(-4)	5.8(-5)	1.5(-4)	1.0(-6)

Table 5.9: Problem 6 Error

t	CN	τ	Stehfest	τ	Talbot	τ
0.5	1.6(-4)	3.7(0)	1.8(-3)	4.9(-2)	8.5(-3)	2.5(-1)
1	3.6(-4)	7.5(0)	1.3(-3)	4.0(-2)	3.2(-3)	2.3(-1)
2	5.1(-5)	1.2(1)	3.0(-4)	4.4(-2)	4.3(-4)	2.3(-1)
3	5.8(-6)	2.2(1)	1.0(-4)	3.6(-2)	5.5(-5)	2.3(-1)
4	1.8(-6)	3.0(1)	6.2(-5)	3.6(-2)	7.8(-6)	1.2(-1)
5	6.6(-7)	3.8(1)	1.5(-4)	3.6(-2)	1.0(-6)	1.3(-1)

Table 5.10: Problem 6 Error $dx = 0.01$

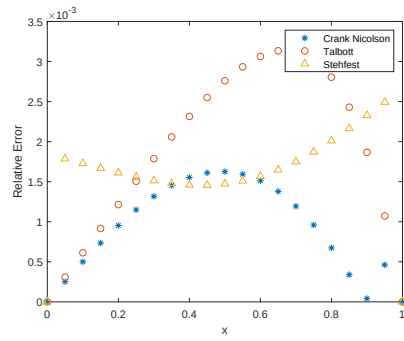


Figure 5.17: $t = 0.5 \delta x = 0.05$

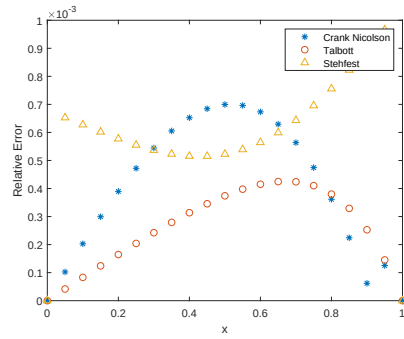


Figure 5.18: $t = 1.0 \delta x = 0.05$.

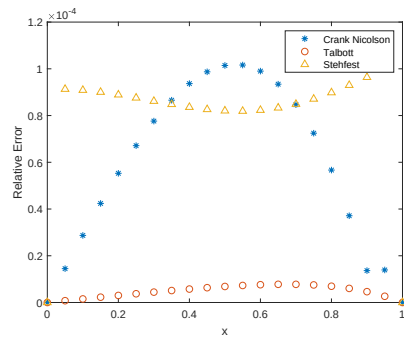


Figure 5.19: $t = 4.0 \delta x = 0.05$.

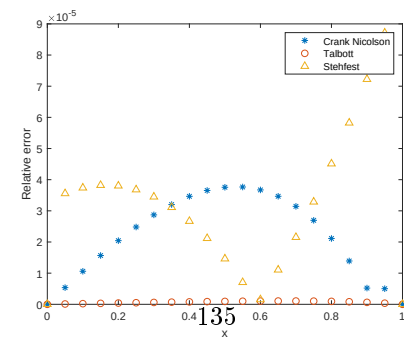


Figure 5.20: $t = 5.0 \delta x = 0.05$.

Problem 6. Table 5.9 shows that for $t = 0.5$, the Crank-Nicholson scheme performs better than the two LTFDM schemes. At $t = 1$, the Crank-Nicholson and Talbot have the same order of error, with the Crank-Nicholson performing marginally better and both doing better than the Stehfest inversion. However, for $t > 2$, the Talbot performs better than both the Crank-Nicholson and the Stehfest inversion methods. Table 5.10 shows all the schemes improving in accuracy for the smaller spatial step $\delta x = 0.01$, but the Crank-Nicholson method does so at the expense of a substantial increase in elapsed times. Figures 5.15 to 5.18 shows the error profile for $\delta x = 0.05$. Here we see that except for $t = 0.5$, the Talbot inversion LTFDM perform better than the other two methods.

5.4.4 Problem 7

The next of this class has the non-linear term e^{-u} . The boundary condition and analytic solution are as in Problem 6.

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t} + e^{-u} + a(x, t) \quad (5.42)$$

with

$$a(x, t) = (2 + x^2)e^{-t} - \exp(-x^2 e^{-t}) \quad (5.43)$$

For this case we express the non-linear term e^{-u^m} in terms of its previously iterated solution, $e^{-u^{m-1}}$ giving

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t} + e^{-u^{m-1}} + (2 + x^2) e^{-t} - \exp(-x^2 e^{-t}) \quad (5.44)$$

However the Laplace transform cannot be performed on the $\exp(-x^2 e^{-t})$ term in (5.44) but we can transform this term into the s -space through its Maclaurin

expansion as follows:

$$\exp(-x^2 e^{-t}) \approx 1 - x^2 e^{-t} + \frac{x^4 e^{-2t}}{2!} - \frac{x^6 e^{-3t}}{3!} + \frac{x^8 e^{-4t}}{4!} \quad (5.45)$$

Now taking the Laplace transform of (5.42) stopping the exponential expansion after the fifth term we have:

$$\frac{\partial^2 \bar{u}}{\partial x^2} = \bar{u} s_j - u(0) + \frac{e^{-u_{m-1}}}{s} + \frac{(2+x^2)}{1+s} - \left(\frac{1}{s} - \frac{x^2}{1+s} + \frac{x^4}{2!(2+s)} - \frac{x^6}{3!(3+s)} + \frac{x^8}{4!(4+s)} \right) \quad (5.46)$$

This gives the finite difference scheme

$$\bar{u}_{i-1}^m - \bar{u}_i^m (2 + \delta x^2) = \bar{u}_{i+1}^m = -\delta x^2 u_i(0) + \delta x^2 \frac{e^{-u_{m-1}}}{s_j} - \delta x^2 \bar{a}(x, s_j) \quad (5.47)$$

Where $\bar{a}(x, s_j)$ is the expression in brackets on the right hand side of (5.46).

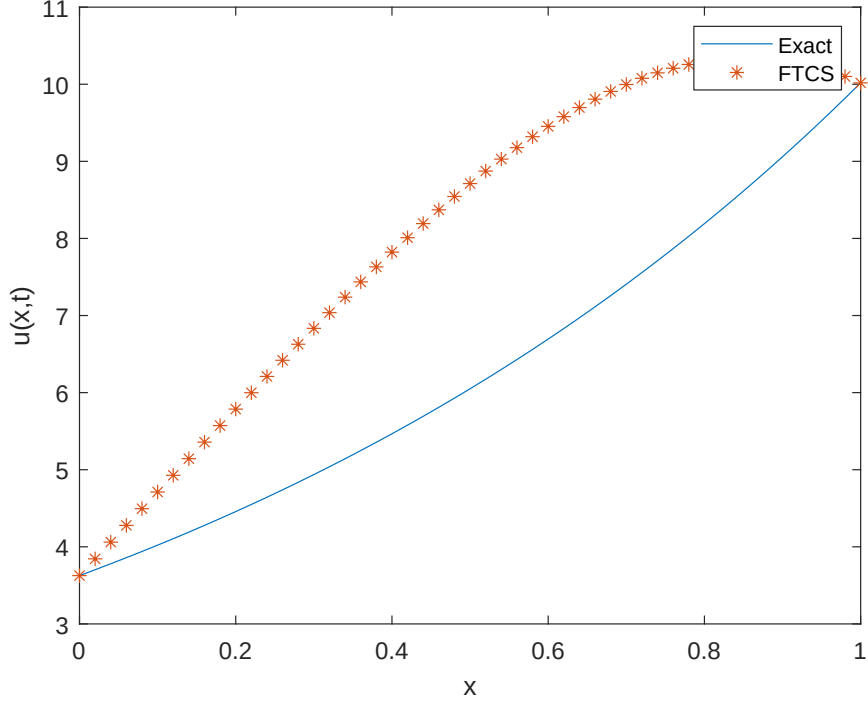


Figure 5.21: FTCS Temperature Profile Problem 7 at $t = 1$.

Figure 5.21 shows that the FTCS method fails to accurately reconstruct the solution for problem 9. so we compare our results with the Crank-Nicholson finite-difference scheme. The Crank Nicholson scheme is

$$ru_{i-1}^{j+1} + ru_i^{j+1}(2+2r) - ru_{i+1}^{j+1} + dt e^{-u_i^{j+1,m}} = ru_{i-1}^j + ru_i^j(2-2r) + ru_{i+1}^j - 2a(x, t) - dt e^{-u_i^{j,m}} \quad (5.48)$$

t	CN	Stehfest	Talbot
0.5	6.0(-3)	1.7(-2)	1.9(-3)
1	3.7(-3)	1.1(-2)	1.8(-4)
2	1.4(-3)	3.9(-3)	3.6(-4)
3	5.3(-4)	1.5(-3)	5.8(-5)
4	2.0(-4)	4.9(-4)	5.6(-5)
5	7.2(-5)	1.5(-4)	1.8(-6)

Table 5.11: Problem 7 Error

t	CN	τ	Stehfest	τ	Talbot	τ
0.5	2.2(-3)	3.0(0)	9.2(-4)	3.1(-2)	1.9(-2)	2.2(-1)
1	1.5(-4)	6.2(0)	2.9(-4)	2.9(-2)	1.8(-4)	1.6(-1)
2	3.7(-4)	1.7(1)	3.0(-4)	3.0(-2)	3.6(-4)	2.2(-1)
3	5.3(-5)	2.7(1)	1.7(-4)	3.0(-1)	5.5(-5)	2.1(-1)
4	2.1(-4)	3.7(1)	6.1(-5)	3.1(-1)	5.7(-5)	2.8(-1)
5	2.9(-5)	8.1(1)	1.5(-4)	3.1(-2)	2.2(-6)	3.4(-1)

Table 5.12: Problem 7 Error $\delta x = 0.01$

The following graphs are with $\delta x = 0.05$

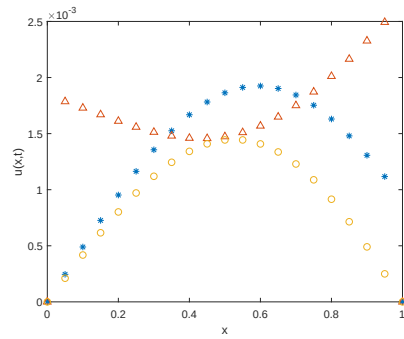


Figure 5.22: $t = 0.5 \delta x = 0.05$

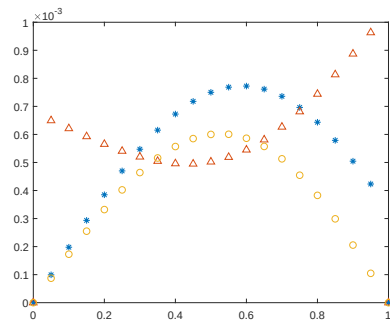


Figure 5.23: $t = 1.0 \delta x = 0.05$

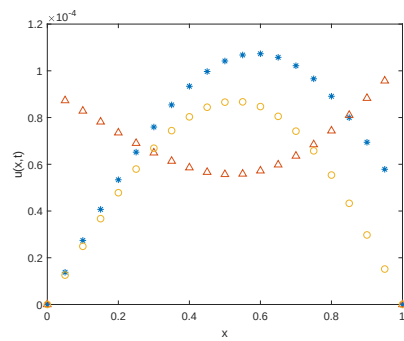


Figure 5.24: $t = 4.0 \delta x = 0.05$

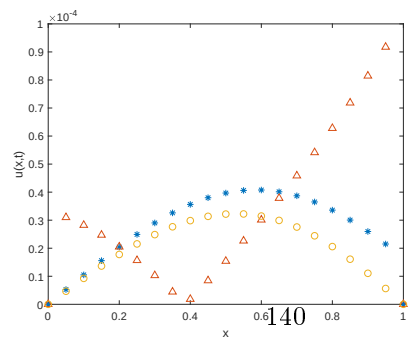


Figure 5.25: $t = 5.0 \delta x = 0.05$ Problem 7.

For this problem, Table 5.11 and Figures 5.19 to 5.22 show that the Talbot LTFDM performs better than the Crank-Nicholson and the Stehfest LTFDM at all times. A decreased spatial step of $\delta x = 0.01$ for both LTFDM schemes provides better results than the Crank-Nicholson method. Again we see that the elapsed times Crank-Nicholson increases significantly as t increases.

5.5 Summary

In this chapter, our tests show that the LTFDM is a viable method for solving linear and nonlinear one-dimensional diffusion equations. The LTFDM, which employed the Talbot inversion scheme, gave better results than the Stehfest inversion scheme for the three linear cases considered in this paper. Moreover, except for early times, the Talbot inversion also performed better than the FTCS in problem two and problem three with considerably shorter elapsed times for both LTFDM schemes in problem 3..

For the nonlinear cases except for problem 4, the LTFDM with Talbot inversion performed better overall than the Stehfest LTFDM and the Crank Nicholson methods. Here except for shorter times, this method produces more accurate results. For example 6, the inability to provide an exact inversion formula for the Laplace Transform at the boundaries may account for the better performance of the FTCS over the LTFDM.

Moreover, while increasing the spatial step δx can lead to improvements in accuracy for all schemes, the Crank-Nicholson method does so with substantial increases in elapsed times when compared with the LTFDM.

We conclude that the LTFDM is a viable alternative to traditional finite-difference schemes for solving one-dimensional diffusion type problems with a particular advantage in accuracy and speed of computation for nonlinear cases. Therefore, these advantages will likely be extended and improved upon for diffusion prob-

lems in higher dimensions.

Chapter 6

Conclusions and Future Work

This thesis examined the use of Laplace Transform Finite Difference method (LTFDM) for solving diffusion type problems. The main drawback in using this method is the problems caused by the perturbation of the numerical inverse transform. We examined the noise handling properties of algorithms from three of the four main classes of Laplace numerical inversion methods. I found that the Talbot inversion algorithm performs with greater accuracy for 'noisy data' than the Fourier Series and Stehfest numerical inversion schemes, as outlined in Chapter 2. This scheme's use of complex arithmetic was the principal reason for its better handling of noisy data. This means that for applications in which such noisy data are likely, I expect the Talbot inversion scheme to perform better and recommend its use.

In Chapter 3, I developed a modified form of the LTFDM to solve the Fisher-KPP reaction-diffusion equation. This equation models the behaviour of a mutant gene in an infinitely long spatial domain. Analysis has shown that the numerical solution suffers from perturbations of an infinite extent. A direct

application of the LTFDM failed to produce accurate results. I concluded that this perturbation was not sufficiently taken into account. Especially so as I was already dealing with a perturbed inversion problem, I concluded that this problem could be overcome by introducing some time discretisation into the process, initialising the initial conditions, and choosing the Talbot algorithm, which is best able to smooth out these perturbations. I compared our results at various wave speeds with a solution generated by perturbation methods. Thus, I could attain results with a maximum percentage error of one per cent. While this means that the advantage of using the Laplace transform to obtain solutions for any time is not fully exploited, the method does allow for considerably larger time steps than is otherwise possible for finite-difference methods.

In Chapter 4, I investigated the performance of five inversion schemes in a multi-precision environment. In standard-double-precision, the Abate-Valko algorithm provides the most accurate results for the numerical reconstructions for the functions tested in this paper. The Fourier algorithm had the worst performance of the five algorithms tested. Both the Stehfest and Salzer-Gaver algorithms had difficulty reconstructing functions of a cyclic nature. None of the algorithms was able to invert the $J_0(t)$ function accurately.

In multi-precision, the Stehfest and the Salzer-Gaver schemes inverted all the functions with high accuracy. The Logan and Abate-Valko schemes could only invert the $J_0(t)$ with limited accuracy. However, they could reconstruct all the other functions with a high degree of accuracy. The most accurate algorithm in multi-precision was the Salzer-Gaver scheme. I also found that the Stehfest algorithm is not merely a Salzer acceleration onto a Gaver functional. However, as Table 4.8 shows, the Salzer-Gaver also had the most extended elapsed times. On the other hand, the Stehfest algorithm had the shortest elapsed times for

the selected functions in Table 4.8. The algorithms that used the Abate-Valko method were the most accurate, but Logan could reconstruct the functions with shorter elapsed times. Therefore I conclude that when working in standard precision, Valko's algorithm performed best. However, in multi-precision, the Stehfest algorithm is best as it inverted all the functions with a high degree (although not the highest) of accuracy and the shortest elapsed times.

In chapter 5, LTFDM and two numerical inversion schemes, Talbot's and Stehfest, were used to successfully solve various linear and nonlinear time-dependent diffusion equations with Dirichlet conditions. The iterative procedures maintained the advantages of solving in the Laplace space. Both inversion schemes provided accurate numerical results for all the equations under consideration.

The LTFDM, which employed the Talbot scheme, gave better results than the Stehfest inversion scheme for the linear cases. Moreover, except for early times, the Talbot inversion also performed better than the FTCS in problem two and problem three, with considerably shorter elapsed times for both LTFDM schemes in problem 3.

For the nonlinear cases except for problem 4, the LTFDM with Talbot inversion performed better overall than the Stehfest LTFDM and the Crank Nicholson methods. Here except for shorter times, this method produces more accurate results. Moreover, while increasing the spatial step δx can improve accuracy for all schemes, the Crank-Nicholson method does so with substantial increases in elapsed times.

I conclude that the LTFDM is a viable alternative to traditional finite-difference schemes for solving one-dimensional diffusion type problems with a particular advantage in accuracy and speed of computation for nonlinear cases. Therefore, these advantages will likely be extended and improved upon for diffusion problems in higher dimensions.

As we have stated in this thesis, the numerical inversion of the Laplace transform is an ill posed problem and hence is a perturbed problem. This means that the solution does not depend continuously on the data, [92]. Therefore, the data for the inverse transform is only known approximately, and convergence from the Laplace s -space back to the time domain is not guaranteed. Hence, some care must be taken when implementing any inversion algorithm, including those outlined in this thesis. One approach uses perturbed data to solve regularised problems to arrive at an accurate solution. The most common is the Tikhonov regularisation scheme,[106], but this requires prior information about the magnitude of the perturbation. In implementing these schemes, one also must consider the trade-offs between accuracy and time constraints, [52]. Therefore, it is recommended that one uses more than one algorithm to invert the Laplace transform numerically. The choice of these algorithms will depend on the problem being solved. This may involve estimations of how much noise is in the data, if the solution has infinite poles on the imaginary axis, if only real data is available, how continuous the data is and so on. I have demonstrated that the Talbot and Stehfest algorithms are both relatively robust for the applications used in the thesis.

6.0.1 Future Work

During the PhD, areas for future investigations arose which could form a basis for future work.

1. I began looking at the possibility of deriving analytical error bounds for the Stehfest algorithm. As Abate and Valko point, [1] out, it is not possible to have such bounds independent of the function, I think it is possible to do so for a particular function. If this is the case, perhaps I could extend this to estimating the errors involved when using the LTFDM with Stehfest inversion for solving

diffusion problems.

2. I derived a new acceleration scheme for the Gaver functional. This produced accurate results for the sine function without the need for multi-precision. However, thus far, I could only invert the exponential and sine functions. This may be more of a coding problem than an issue with the algorithm. Further investigation is needed to determine if this could become a viable method for numerically inverting the Laplace Transform .

3. I have had some success with the LTFDM providing accurate results for the travelling wave solutions of the Fisher-KPP equation. I want to investigate whether this method can be successfully applied to other PDE's which exhibit travelling wave solutions. Also, it might be worth investigating the performance of Laplace Transform methods in solving the wave equation.

4. I would like to systematically investigate the various methods of linearising nonlinear diffusion equations in the context of the LTFDM or, more generally, for methods that use Laplace transform inversion to solve the diffusion equation.

5. I would like to investigate further using the accelerated Fourier series method. While this is not a one-parameter class, as is the case for both the Stehfest and Talbot schemes, it would be interesting to see if there are conditions to make it so. Then this would allow us to perform systematic testing of the algorithm in this form for applications to the LTFDM.

Chapter 7

References

Bibliography

- [1] ABATE, J., AND VALKÓ, P. P. Multi-precision laplace transform inversion. *International Journal for Numerical Methods in Engineering* 60, 5 (2004), 979–993.
- [2] ABLOWITZ, M. J., AND ZEPPETELLA, A. Explicit solutions of fisher’s equation for a special wave speed. *Bulletin of Mathematical Biology* 41, 6 (1979), 835–840.
- [3] AHN, J., KANG, S., AND KWON, Y. A flexible inverse laplace transform algorithm and its application. *Computing* 71, 2 (2003), 115–131.
- [4] AHN, J., KANG, S., AND KWON, Y. A laplace transform finite difference method for the black–scholes equation. *Mathematical and computer modelling* 51, 3-4 (2010), 247–255.
- [5] AKGÜL, A., AND MODANLI, M. Crank–nicholson difference method and reproducing kernel function for third order fractional differential equations in the sense of atangana–baleanu caputo derivative. *Chaos, Solitons & Fractals* 127 (2019), 10–16.
- [6] AMBROSIO, L., MONDINO, A., AND SAVARÉ, G. *Nonlinear diffusion equations and curvature conditions in metric measure spaces*, vol. 262. American Mathematical Society, 2019.

- [7] AWAIS, M. Applications of the numerical inversion of the laplace transform to unsteady problems of the third grade fluid. *Applied mathematics and computation* 250 (2015), 228–234.
- [8] BAKHOUM, E. G., AND TOMA, C. Mathematical transform of traveling-wave equations and phase aspects of quantum interaction. *Mathematical Problems in Engineering* 2010 (2010).
- [9] BATHE, K.-J. Finite element method. *Wiley encyclopedia of computer science and engineering* (2007), 1–12.
- [10] BELLMAN, R., AND ROTH, R. S. *The laplace transform*, vol. 3. World Scientific, 1984.
- [11] BENDER, C. M., AND ORSZAG, S. A. *Advanced mathematical methods for scientists and engineers I: Asymptotic methods and perturbation theory*. Springer Science & Business Media, 2013.
- [12] BERG, J., AND NYSTRÖM, K. A unified deep artificial neural network approach to partial differential equations in complex geometries. *Neurocomputing* 317 (2018), 28–41.
- [13] BLACK, F., AND SCHOLES, M. The pricing of options and corporate liabilities. In *World Scientific Reference on Contingent Claims Analysis in Corporate Finance: Volume 1: Foundations of CCA and Equity Valuation*. World Scientific, 2019, pp. 3–21.
- [14] BONNET, M. *Boundary integral equation methods for solids and fluids*. No. BOOK. John Wiley, 1995.
- [15] BREBBIA, C. A., TELLES, J. C. F., AND WROBEL, L. C. *Boundary element techniques: theory and applications in engineering*. Springer Science & Business Media, 2012.

- [16] BRUSH, S. G. Joseph fourier: The man and the physicist: By john herivel. oxford (clarendon press). 1975. xii+ 350 pp. 30.95 *Historia Mathematica* 4, 2 (1977), 219–221.
- [17] CANOSA, J. On a nonlinear diffusion equation describing population growth. *IBM Journal of Research and Development* 17, 4 (1973), 307–313.
- [18] CAO, L. *Numerical analysis and multi-precision computational methods applied to the extant problems of Asian option pricing and simulating stable distributions and unit root densities*. PhD thesis, University of St Andrews, 2014.
- [19] CATTANI, C., AND KUDREYKO, A. Mutiscale analysis of the fisher equation. In *International Conference on Computational Science and Its Applications* (2008), Springer, pp. 1171–1180.
- [20] CHEN, H.-T., AND LIN, J.-Y. Application of the laplace transform to nonlinear transient problems. *Applied mathematical modelling* 15, 3 (1991), 144–151.
- [21] CHEN, J.-S., HILLMAN, M., AND CHI, S.-W. Meshfree methods: progress made after 20 years. *Journal of Engineering Mechanics* 143, 4 (2017), 04017001.
- [22] CHENG, A. H.-D., AND CHENG, D. T. Heritage and early history of the boundary element method. *Engineering analysis with boundary elements* 29, 3 (2005), 268–302.
- [23] COHEN, A. M. *Numerical methods for Laplace transform inversion*, vol. 5. Springer Science & Business Media, 2007.

- [24] COOLEY, J. W., LEWIS, P., AND WELCH, P. The fast fourier transform algorithm: Programming considerations in the calculation of sine, cosine and laplace transforms. *Journal of Sound and Vibration* 12, 3 (1970), 315–337.
- [25] COOLEY, J. W., AND TUKEY, J. W. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation* 19, 90 (1965), 297–301.
- [26] CRAIG, I., THOMPSON, A., AND THOMPSON, W. J. Practical numerical algorithms why laplace transforms are difficult to invert numerically. *Computers in Physics* 8, 6 (1994), 648–653.
- [27] CRANN, D. The laplace transform boundary element method for diffusion-type problems.
- [28] CRUMP, K. S. Numerical inversion of laplace transforms using a fourier series approximation. *Journal of the ACM (JACM)* 23, 1 (1976), 89–96.
- [29] CURRAN, D., CROSS, M., AND LEWIS, B. Solution of parabolic differential equations by the boundary element method using discretisation in time. *Applied Mathematical Modelling* 4, 5 (1980), 398–400.
- [30] DARDAGANIAN, S. G., ET AL. The application of the buckley-leverett frontal advance theory to petroleum recovery. *Journal of petroleum technology* 10, 04 (1958), 49–52.
- [31] DAVIES, A. Using a spreadsheet to investigate models of heat conduction and vibrating strings. *Teaching Mathematics and its Applications: An International Journal of the IMA* 12, 4 (1993), 179–185.
- [32] DAVIES, A., AND CRANN, D. *A handbook of essential mathematical formulae*. Univ of Hertfordshire Press, 2004.

- [33] DAVIES, A., CRANN, D., KANE, S., AND LAI, C.-H. A hybrid laplace transform/finite difference boundary element method for diffusion problems. *Computer Modelling in Engineering and Sciences* 18, 2 (2007), 79–86.
- [34] DAVIES, A., CRANN, D., AND MUSHTAQ, J. A parallel implementation of the laplace transform bem. boundary elements xviii, eds. brebbia ca, martins jb, aliabadi mh and haie n, 213-222, 1996.
- [35] DAVIES, B. *Integral transforms and their applications*, vol. 41. Springer Science & Business Media, 2002.
- [36] DAVIES, B., AND MARTIN, B. Numerical inversion of the laplace transform: a survey and comparison of methods. *Journal of computational physics* 33, 1 (1979), 1–32.
- [37] DE HOOG, F. R., KNIGHT, J., AND STOKES, A. An improved method for numerical inversion of laplace transforms. *SIAM Journal on Scientific and Statistical Computing* 3, 3 (1982), 357–366.
- [38] DEFREITAS, C. L., AND KANE, S. J. The noise handling properties of the talbot algorithm for numerically inverting the laplace transform. *Journal of Algorithms & Computational Technology* 13 (2018), 1748301818797069.
- [39] DUBNER, H., AND ABATE, J. Numerical inversion of laplace transforms by relating them to the finite fourier cosine transform. *Journal of the ACM (JACM)* 15, 1 (1968), 115–123.
- [40] DUFFY, D. G. On the numerical inversion of laplace transforms: comparison of three new methods on characteristic problems from applications. *ACM Transactions on Mathematical Software (TOMS)* 19, 3 (1993), 333–359.

- [41] DUFFY, D. G. *Transform methods for solving partial differential equations*. Chapman and Hall/CRC, 2019.
- [42] DURBIN, F. Numerical inversion of laplace transforms: an efficient improvement to dubner and abate’s method. *The Computer Journal* 17, 4 (1974), 371–376.
- [43] EGONMWAN, A. The numerical inversion of the laplace transform: Gaver-stehfest. *Piessens, and Regularized Collocation methods, LAP LAMBERT Academic Publishing: Saarbrücken, Germany* (2012).
- [44] EL-AJOU, A., ARQUB, O. A., MOMANI, S., BALEANU, D., AND AL-SAEDI, A. A novel expansion iterative method for solving linear partial differential equations of fractional order. *Applied Mathematics and Computation* 257 (2015), 119–133.
- [45] EPSTEIN, C. L., AND SCHOTLAND, J. The bad truth about laplace’s transform. *SIAM review* 50, 3 (2008), 504–520.
- [46] EYMARD, R., GALLOUËT, T., AND HERBIN, R. Finite volume methods. *Handbook of numerical analysis* 7 (2000), 713–1018.
- [47] FATOOREHCHI, H., AND ABOLGHAEMI, H. Series solution of nonlinear differential equations by a novel extension of the laplace transform method. *International Journal of Computer Mathematics* 93, 8 (2016), 1299–1319.
- [48] FISHER, R. A. The wave of advance of advantageous genes. *Annals of eugenics* 7, 4 (1937), 355–369.
- [49] FITZHARRIS, A. Parallel solution of diffusion equations using laplace transform methods with particular reference to black-scholes models of financial options.

- [50] GAVER JR, D. P. Observing stochastic processes, and approximate transform inversion. *Operations Research* 14, 3 (1966), 444–459.
- [51] GAZDAG, J., AND CANOSA, J. Numerical solution of fisher’s equation. *Journal of Applied Probability* 11, 3 (1974), 445–457.
- [52] GIRYES, R., ELDAR, Y. C., BRONSTEIN, A. M., AND SAPIRO, G. Trade-offs between convergence speed and reconstruction accuracy in inverse problems. *IEEE Transactions on Signal Processing* 66, 7 (2018), 1676–1690.
- [53] GRIEBEL, M., SCHWEITZER, M. A., ET AL. *Meshfree methods for partial differential equations II*. Springer, 2005.
- [54] HABTE, A. D., ONUR, M., ET AL. Laplace-transform finite-difference and quasistationary solution method for water-injection/falloff tests. *SPE Journal* 19, 03 (2014), 398–409.
- [55] HONIG, G., AND HIRDES, U. A method for the numerical inversion of laplace transforms. *Journal of Computational and Applied Mathematics* 10, 1 (1984), 113–132.
- [56] HONNOR, M., AND DAVIES, A. Nonlinear transient field problems with phase change using the boundary element method. *Engineering analysis with boundary elements* 28, 6 (2004), 561–570.
- [57] JACOBS, B. A. High-order compact finite difference and laplace transform method for the solution of time-fractional heat equations with dirchlet and neumann boundary conditions. *Numerical Methods for Partial Differential Equations* 32, 4 (2016), 1184–1199.

- [58] KANE, S., DAVIES, A., CRANN, D., AND LAI, C.-H. A hybrid laplace transform/finite difference boundary element method for diffusion problems. *Computer Modelling in Engineering and Sciences* (2007).
- [59] KROUGLY, Z., DAVISON, M., AIYAR, S., ET AL. The role of high precision arithmetic in calculating numerical laplace and inverse laplace transforms. *Applied Mathematics* 8, 04 (2017), 562.
- [60] KUHLMAN, K. L. Comparison of inverse laplace transform algorithms for laplace-space numerical approaches. Tech. rep., Sandia National Laboratories, 2012.
- [61] KUHLMAN, K. L. Review of inverse laplace transform algorithms for laplace-space numerical approaches. *Numerical Algorithms* 63, 2 (2013), 339–355.
- [62] KUZNETSOV, A. On the convergence of the gaver–stehfest algorithm. *SIAM Journal on Numerical Analysis* 51, 6 (2013), 2984–2998.
- [63] LAM, C.-Y. *Applied numerical methods for partial differential equations: an introduction with spreadsheet programs*. Prentice-Hall, 1994.
- [64] LAVRENT_EV, M. M., ROMANOV, V. G., AND SHISHATSKI_, S. P. *Ill-posed problems of mathematical physics and analysis*, vol. 64. American Mathematical Soc., 1986.
- [65] LI, J., FARQUHARSON, C. G., AND HU, X. Three effective inverse laplace transform algorithms for computing time-domain electromagnetic responses. *Geophysics* 81, 2 (2016), E113–E128.
- [66] LOGAN, J. D. *Transport modeling in hydrogeochemical systems*, vol. 15. Springer Science & Business Media, 2013.

- [67] MA, J., ZHOU, Z., AND CUI, Z. Hybrid laplace transform and finite difference methods for pricing american options under complex models. *Computers & Mathematics with Applications* 74, 3 (2017), 369–384.
- [68] MAHAJERIN, E., AND BURGESS, G. A laplace transform-based fundamental collocation method for two-dimensional transient heat flow. *Applied thermal engineering* 23, 1 (2003), 101–111.
- [69] MARTÍN-VAQUERO, J., AND SAJAVIČIUS, S. The two-level finite difference schemes for the heat equation with nonlocal initial condition. *Applied Mathematics and Computation* 342 (2019), 166–177.
- [70] MCWHIRTER, J., AND PIKE, E. R. On the numerical inversion of the laplace transform and similar fredholm integral equations of the first kind. *Journal of Physics A: Mathematical and General* 11, 9 (1978), 1729.
- [71] MIRZAEI, F., AND SAMADYAR, N. Combination of finite difference method and meshless method based on radial basis functions to solve fractional stochastic advection–diffusion equations. *Engineering with computers* 36, 4 (2020), 1673–1686.
- [72] MORAN, P. A. P., ET AL. The statistical processes of evolutionary theory. *The statistical processes of evolutionary theory*. (1962).
- [73] MORIDIS, G. Alternative formulations of the laplace transform boundary element (ltbe) numerical method for the solution of diffusion-type equations. In *Boundary element technology VII*. Springer, 1992, pp. 815–833.
- [74] MORIDIS, G. J., AND REDDELL, D. L. The laplace transform finite difference method for simulation of flow through porous media. *Water Resources Research* 27, 8 (1991), 1873–1884.

- [75] MURLI, A., AND RIZZARDI, M. Algorithm 682: Talbot's method of the laplace inversion problems. *ACM Transactions on Mathematical Software (TOMS)* 16, 2 (1990), 158–168.
- [76] MURRAY, J. Mathematical biology 1: An introduction, ser. interdisciplinary applied sciences, 2002.
- [77] MURRAY, J. D. *Mathematical biology: I. An introduction*, vol. 17. Springer Science & Business Media, 2007.
- [78] NARAYANAN, G., AND BESKOS, D. Numerical operational methods for time-dependent linear problems. *International Journal for Numerical Methods in Engineering* 18, 12 (1982), 1829–1854.
- [79] OBERHETTINGER, F., AND BADI, L. *Tables of Laplace transforms*. Springer Science & Business Media, 2012.
- [80] OLVER, P. J. *Introduction to partial differential equations*. Springer, 2014.
- [81] OPANUGA, A., OWOLOKO, E., OKAGBUE, H., AND AGBOOLA, O. Finite difference method and laplace transform for boundary value problems.
- [82] ÖZİŞİK, M. N., ORLANDE, H. R., COLACO, M. J., AND COTTA, R. M. *Finite difference methods in heat transfer*. CRC press, 2017.
- [83] PHILLIPS, R. S. An inversion formula for laplace transforms and semi-groups of linear operators. *Annals of Mathematics* (1954), 325–356.
- [84] PIESSENS, R. A new numerical method for the inversion of the laplace transform. *IMA Journal of Applied Mathematics* 10, 2 (1972), 185–192.
- [85] POLYANIN, A. D., SCHIESSER, W. E., AND ZHUROV, A. I. Partial differential equation. *Scholarpedia* 3, 10 (2008), 4605.

- [86] PRAKASH, A., AND KUMAR, M. Numerical method for solving time-fractional multi-dimensional diffusion equations. *International Journal of Computing Science and Mathematics* 8, 3 (2017), 257–267.
- [87] RADFORD, L. E. *Aspects of the Laplace transform isotherm migration method*. PhD thesis, 2008.
- [88] RANI, D., AND MISHRA, V. Numerical inverse laplace transform based on bernoulli polynomials operational matrix for solving nonlinear differential equations. *Results in Physics* 16 (2020), 102836.
- [89] RANI, D., MISHRA, V., AND CATTANI, C. Numerical inversion of laplace transform based on bernstein operational matrix. *Mathematical Methods in the Applied Sciences* 41, 18 (2018), 9231–9243.
- [90] RANI, D., MISHRA, V., AND CATTANI, C. Numerical inverse laplace transform for solving a class of fractional differential equations. *Symmetry* 11, 4 (2019), 530.
- [91] SCHAPERY, R. Two simple approximate methods of laplace transform inversion for viscoelastic stress analysis.
- [92] SEIDMAN, T. I., AND VOGEL, C. R. Well posedness and convergence of some regularisation methods for non-linear ill posed problems. *Inverse problems* 5, 2 (1989), 227.
- [93] SIMMONDS, J. G., AND MANN, J. E. *A first look at perturbation theory*. Courier Corporation, 2013.
- [94] SIMON, R. M., STROOT, M. T., AND WEISS, G. H. Numerical inversion of laplace transforms with application to percentage labeled mitoses experiments. *Computers and Biomedical Research* 5, 6 (1972), 596–607.

- [95] SINGH, K. M., AND KALRA, M. S. Time integration in the dual reciprocity boundary element analysis of transient diffusion. *Engineering analysis with boundary elements* 18, 2 (1996), 73–102.
- [96] SMITH, G. D., SMITH, G. D., AND SMITH, G. D. S. *Numerical solution of partial differential equations: finite difference methods*. Oxford university press, 1985.
- [97] SPIEGEL, M. R. *Laplace transforms*. McGraw-Hill New York, 1965.
- [98] SPIEGEL, M. R. *Schaum's outline of theory and problems of Laplace transforms*. McGraw-Hill, 1965.
- [99] STEHFEST, H. Algorithm 368: Numerical inversion of laplace transforms [d5]. *Communications of the ACM* 13, 1 (1970), 47–49.
- [100] SUN, G., AND TRUEMAN, C. W. Efficient implementations of the crank-nicolson scheme for the finite-difference time-domain method. *IEEE transactions on microwave theory and techniques* 54, 5 (2006), 2275–2284.
- [101] SUTRADHAR, A., PAULINO, G. H., AND GRAY, L. Transient heat conduction in homogeneous and non-homogeneous materials by the laplace transform galerkin boundary element method. *Engineering Analysis with Boundary Elements* 26, 2 (2002), 119–132.
- [102] SVÄRD, M., AND NORDSTRÖM, J. On the convergence rates of energy-stable finite-difference schemes. *Journal of Computational Physics* 397 (2019), 108819.
- [103] TAGLIANI, A., AND MILEV, M. Laplace transform and finite difference methods for the black-scholes equation. *Applied Mathematics and Computation* 220 (2013), 649–658.

- [104] TALBOT, A. The accurate numerical inversion of laplace transforms. *IMA Journal of Applied Mathematics* 23, 1 (1979), 97–120.
- [105] THOMAS, J. W. *Numerical partial differential equations: finite difference methods*, vol. 22. Springer Science & Business Media, 2013.
- [106] TIKHONOV, A. N., AND ARSENIN, V. Y. Solutions of ill-posed problems. vh winston & sons, 1977.
- [107] VERMA, A., JIWARI, R., AND KOKSAL, M. E. Analytic and numerical solutions of nonlinear diffusion equations via symmetry reductions. *Advances in Difference Equations* 2014, 1 (2014), 229.
- [108] VERSTEEG, H. K., AND MALALASEKERA, W. *An introduction to computational fluid dynamics: the finite volume method*. Pearson education, 2007.
- [109] WANG, Q., AND ZHAN, H. On different numerical inverse laplace methods for solute transport problems. *Advances in Water Resources* 75 (2015), 80–92.
- [110] WEIDEMAN, J., AND TREFETHEN, L. Parabolic and hyperbolic contours for computing the bromwich integral. *Mathematics of Computation* 76, 259 (2007), 1341–1356.
- [111] WEIDEMAN, J. A. C. Algorithms for parameter selection in the weeks method for inverting the laplace transform. *SIAM Journal on Scientific Computing* 21, 1 (1999), 111–128.
- [112] WIMP, J., ET AL. Sequence transformations and their applications.
- [113] XI, Q., CHEN, C., FU, Z., AND COMINO, E. The maps with polynomial basis functions for solving axisymmetric time-fractional equations. *Computers & Mathematics with Applications* (2019).

- [114] YANG, X., AND DANG, X. A new parallel difference algorithm based on improved alternating segment crank–nicolson scheme for time fractional reaction–diffusion equation. *Advances in Difference Equations* 2019, 1 (2019), 1–18.
- [115] YANG, X.-J., AND GAO, F. A new technology for solving diffusion and heat equations. *Thermal Science* 21, 1 Part A (2017), 133–140.
- [116] ZAHRA, W., HIKAL, M., AND BAHNASY, T. A. Solutions of fractional order electrical circuits via laplace transform and nonstandard finite difference method. *Journal of the Egyptian Mathematical Society* 25, 2 (2017), 252–261.
- [117] ZHU, S., AND SATRAVAHA, P. Combined laplace transform and dual reciprocity method for solving time-dependent diffusion equations with nonlinear source terms. *WIT Transactions on Modelling and Simulation* 7 (1970).
- [118] ZHU, S., AND SATRAVAHA, P. An efficient computational method for modelling transient heat conduction with nonlinear source terms. *Applied mathematical modelling* 20, 7 (1996), 513–522.
- [119] ZHU, S., AND SATRAVAHA, P. Solving nonlinear time-dependent diffusion equations with the dual reciprocity method in laplace space. *Engineering analysis with boundary elements* 18, 1 (1996), 19–27.