UNIVERSITY OF THE
WITWATERSRAND,
JOHANNESBURG

University of Witwatersrand
Johannesburg, South Africa

APPM3017 Numerical Methods Assignment

Project 5

Group 11: Kimessha Paupamah, Tamlin Love, Sakhile Mbane, Tshepo Tsilo, Mwamba Nankonde

1038238, 1438243, 1042627, 941431, 1453499

Application of the Finite Difference and Non-Standard Finite Difference Methods to the Fisher Equation

---

Semester 2, 2018

Abstract

We investigate the performance of two numerical schemes - the Explicit Finite Difference and Non-Standard Finite Difference - in solving the Fisher Equation $\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \rho u(1-u)$, a Reaction-Diffusion class PDE used in a number of biological, ecological and chemical models. We model the two schemes for varying step sizes in x and t and compare each to the exact analytical solution, $\frac{1}{(\exp(\sqrt{\frac{\rho}{6}}x - \frac{5\rho}{6}t)+1)^2}$. We conclude that the Non-Standard scheme is overall more robust than the Finite Difference scheme, given its performance with large Δx and Δt values.

1 Introduction

Non-linear Partial Differential Equations are used to model many phenomena in science and industry, including fluid mechanics, optics, quantum field theory and population genetics. In general, these equations are difficult, or even impossible, to solve analytically, thus necessitating the use of numerical methods.
We will be examining a particular class of partial differential equations, the Reaction-Diffusion Equations. From [1], we note that positivity is preserved for this class of equations. In particular, we will focus on the Fisher Equation (1), also known as the Kolmogorov–Petrovsky–Piskunov Equation, which was first introduced by Ronald Fisher in modelling the spread of advantageous genes in a population [4].

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

Subject to

$$u(x,0) = \frac{1}{(e^{10(\frac{10}{3})^{0.5}x} + 1)^2}$$

and Dirichlet boundary conditions

$$u(-1,t) = 1, u(1,t) = 0$$

or Neumann boundary conditions

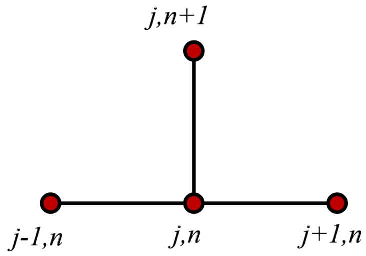
$$u_x(0,t) = 0, u_x(1,t) = 0$$

with

$$\rho = 2000$$

In this paper we consider two approaches, the Finite Difference and Non-Standard Finite Difference approaches, both of which operate on the principle of approximating the continuous system as a mesh of discrete points. We consider the explicit forms of these methods, meaning that the function value of $u(x,t)$ for a single point at time $t + \Delta t$ is determined by the values of three neighbouring points at time t (see Figure 1). The usefulness of these schemes depend greatly on the choice of temporal and spatial step sizes, investigated at length below. The Non-Standard Finite Difference scheme, in particular, is designed to address the numerical instabilities inherent to such schemes [1, 2].

Figure 1: The Explicit Stencil which approximates the continuous time and space as a discrete mesh



In Sections 2 and 3 we formally introduce these schemes and analyse their consistency, stability and convergence. In Section 4 we verify our theoretical analysis through experimental methods, the consequences of which are discussed in Section 5.

2 Finite Difference Approach

The Finite Difference method works, as stated in the introduction, by approximating the continuous space and time as a discrete grid of points. Thus the dependent variable $u(x_i, t_n)$ can be reformulated as u_i^n, where i denotes a position in space and n denotes a moment in time.
Since the explicit form of the method is used, u_i^{n+1} is determined by some combination of u_{i-1}^n, u_i^n and u_{i+1}^n, derived below.

2.1 Methodology

. We begin with (1),

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

Using the Taylor Series expansion, we approximate the temporal derivative with a first order discretised approximation

$$u_t(x_i, t_n) = \frac{u(x_i, t_{n+1}) - u(x_i, t_n)}{\Delta t} + O(\Delta t) = \frac{u_i^{n+1} - u_i^n}{\Delta t} + O(\Delta t)$$

Similarly, we approximate the second order spatial derivative with a second order discretised approximation

$$\begin{aligned} u_{xx}(x_i, t_n) &= \frac{u(x_{i+1}, t_n) - 2u(x_i, t_n) + u(x_{i-1}, t_n)}{\Delta x^2} + O(\Delta x^2) \\ &= \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} + O(\Delta x^2) \end{aligned}$$

Dropping the truncation errors, we obtain

$$u_t(x_i, t_n) = \frac{u_i^{n+1} - u_i^n}{\Delta t} \tag{2}$$

$$u_{xx}(x_i, t_n) = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} \tag{3}$$

Inserting equations (2) and (3) into equation (1), we arrive at

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} + \rho u_i^n(1 - u_i^n) \tag{4}$$

Which can be rearranged to arrive at our scheme

$$u_i^{n+1} = u_i^n + R(u_{i+1}^n - 2u_i^n + u_{i-1}^n) + \rho \Delta t u_i^n(1 - u_i^n) \tag{5}$$

where $R = \frac{\Delta t}{\Delta x^2}$

2.2 Analysis

To prove the efficacy and usefulness of the approach, we prove consistency, stability (via von Neumann Stability Analysis) and convergence.

2.2.1 Consistency

To show consistency, we begin with (4), which of course is just a rearrangement of the scheme derived above.

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} + \rho u_i^n(1 - u_i^n)$$

Taking the limits to 0 of the step sizes Δx and Δt

$$\begin{aligned}\lim_{\Delta x \to 0, \Delta t \to 0} \frac{u_i^{n+1} - u_i^n}{\Delta t} &= \lim_{\Delta x \to 0, \Delta t \to 0} \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} \\ &+ \lim_{\Delta x \to 0, \Delta t \to 0} \rho u_i^n(1 - u_i^n)\end{aligned}$$

Which, from the Taylor Series, recovers (1),

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

$\square$

2.2.2 Stability

. Consider our scheme from (5).

$$u_i^{n+1} = u_i^n + R(u_{i+1}^n - 2u_i^n + u_{i-1}^n) + \rho \Delta t u_i^n(1 - u_i^n)$$

We assume an Ansatz in the form of a Fourier mode

$$u_i^n = \xi^n e^{Iqi\Delta x}$$

where q is a real spatial wave number of the Fourier series , $I = (-1)^{0.5}$ and i and n are the usual indices. We linearise the scheme by replacing the $u_i^n(1-u_i^n)$ with $u_i^n(1 - C)$, where C is constant.

$$u_i^{n+1} = u_i^n + R(u_{i+1}^n - 2u_i^n + u_{i-1}^n) + \rho \Delta t u_i^n(1 - C)$$

Substituting this Ansatz into our numerical scheme, we arrive at

$$\begin{aligned}\xi^{n+1}e^{Iqi\Delta x} &= \xi^n e^{Iqi\Delta x} + R\xi^n(e^{Iq(i+1)\Delta x} - 2e^{Iqi\Delta x} + e^{Iq(i-1)\Delta x}) \\ &+ \rho\Delta t\xi^n e^{Iqi\Delta x}(1-C)\end{aligned}$$

Dividing through by $\xi^n e^{Iqi\Delta x}$ we get

$$\xi = 1 + R(e^{Iq\Delta x} - 2 + e^{-Iq\Delta x}) + \rho\Delta t(1-C))$$

$$\implies \xi = 1 + 2R(\frac{e^{Iq\Delta x} + e^{-Iq\Delta x}}{2} - 1) + \rho\Delta x(1-C)$$

Now using Euler's formula and the double angle formula for $\cos(\theta)$ we get,

$$\xi = 1 - 4R(\sin^2(\frac{q\Delta x}{2})) + \rho\Delta t(1-C)$$

for our scheme to be stable, we need the amplification factor $-1 < \xi < 1$. The conditions we need for Δx and Δt such that $-1 < \xi < 1$ holds is,

$$-1 < 1 - 4R(\sin^2(\frac{q\Delta x}{2})) + \rho\Delta t(1-C) < 1$$

Equivalently,

$$0 < 2 - 4R(\sin^2(\frac{q\Delta x}{2})) + \rho\Delta t(1-C) < 2 \tag{6}$$

Now taking the RHS of equation (6) we have that

$$\implies 2 - 4R(\sin^2(\frac{q\Delta x}{2})) + \rho\Delta t(1-C) < 2$$

$$\implies \rho\Delta t(1-C) < 4R\sin^2(\frac{q\Delta x}{2})$$

We note that $\max(\sin^2(\frac{q\Delta x}{2})) = 1$. Thus

$$\implies \rho\Delta t(1-C) < 4R$$

We choose $C = 0$ in order to maximise $(1 - u_i^n)$ for $u_i^n > 0$

$$\rho\Delta t < 4R$$

$$\implies R > \frac{\rho\Delta t}{4}$$

Now consider the LHS of equation (6)

$$-1 < 1 - 4R\sin^2(\frac{q\Delta x}{2}) + \rho\Delta t(1-C)$$

Since $max(\sin^2(\frac{q\Delta x}{2})) = 1$ and $C = 0$, we obtain

$$4R < 2 + \rho\Delta t$$

$$\implies R < \frac{2 + \rho\Delta t}{4}$$

Thus the Finite Difference scheme is conditionally stable for $\frac{\rho\Delta t}{4} < R < \frac{2+\rho\Delta t}{4}$. Expressed in terms of Δx and Δt

$$\frac{4\Delta t}{2 + \rho\Delta t} < \Delta x^2 < \frac{4}{\rho} \tag{7}$$

□

2.2.3 Convergence

. By the Lax Equivalence Theorem, the scheme is convergent. □

3 Non-Standard Finite Difference Approach

We now examine the Non-Standard Finite Difference approach, Introduced by R.E. Mickens. [1].

3.1 Methodology

From 1 we have

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \rho u - \rho u^2$$

We know that a finite difference scheme is considered non-standard if the non-linear term ρu^2 is replaced by a non-local discretisation [3]. Using the non-local discretisation in [1], we obtain

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} + \rho u_i^n - \rho(\frac{u_{i+1}^n + u_i^n + u_{i-1}^n}{3})u_i^{n+1} \tag{8}$$

with u^2 being replaced by the non-local discretisation $(\frac{u_{i+1}^n + u_i^n + u_{i-1}^n}{3})u_i^{n+1}$.

Solving for u_i^{n+1} we get

$$u_i^{n+1} = \frac{R(u_{i+1}^n + u_{i-1}^n) + u_i^n(1 + \rho\Delta t - 2R)}{1 + (\frac{\rho\Delta t}{3})(u_{i+1}^n + u_i^n + u_{i-1}^n)} \tag{9}$$

where $R = \frac{\Delta t}{\Delta x^2}$. This gives us our non-standard finite difference scheme.

3.2 Analysis

As with the Finite Difference method, we prove consistency, stability (via von Neumann Stability Analysis) and convergence.

3.2.1 Consistency

. From (8) we get

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} + \rho u_i^n - \rho(\frac{u_{i+1}^n + u_i^n + u_{i-1}^n}{3})u_i^{n+1}$$

Applying limits,

$$\begin{aligned}\lim_{\Delta x, \Delta t \to 0} \frac{u_i^{n+1} - u_i^n}{\Delta t} = &\lim_{\Delta x, \Delta t \to 0} \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} + \lim_{\Delta x, \Delta t \to 0} \rho u_i^n \\ &- \lim_{\Delta x, \Delta t \to 0} \rho(\frac{u_{i+1}^n + u_i^n + u_{i-1}^n}{3})u_i^{n+1}\end{aligned}$$

which, from Taylor series, recovers (1),

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \rho u - \rho u^2$$

□

3.2.2 Stability

. Rewriting our scheme (9), we obtain

$$u_i^{n+1} + \frac{\rho\Delta t}{3}(u_{i+1}^n + u_i^n + u_{i-1}^n)u_i^{n+1} = R(u_{i+1}^n - 2u_i^n + u_{i-1}^n) + \rho\Delta t u_i^n + u_i^n$$

We assume an Ansatz in the form of a Fourier mode

$$u_i^n = \xi^n e^{Iqi\Delta x}$$

where q is a real spatial wave number of the Fourier series , $I = (-1)^{0.5}$ and i and n are the usual indices. We linearise the scheme by replacing the term $\frac{\rho\Delta t}{3}(u_{i+1}^n + u_i^n + u_{i-1}^n)u_i^{n+1}$ with $\frac{\rho\Delta t}{3}(C)u_i^{n+1}$, where C is a constant, and get

$$u_i^{n+1} + C\frac{\rho\Delta t}{3}u_i^{n+1} = R(u_{i+1}^n - 2u_i^n + u_{i-1}^n) + \rho\Delta t u_i^n + u_i^n$$

Substituting this Ansatz into our numerical scheme, we arrive at

$$\begin{aligned}\xi^{n+1}e^{Iqi\Delta x} + C\frac{\rho\Delta t}{3}\xi^{n+1}e^{Iqi\Delta x} &= R\xi^n(e^{Iq(i+1)\Delta x} - 2e^{Iqi\Delta x} + e^{Iq(i-1)\Delta x}) \\ &\quad + \rho\Delta t\xi^n e^{Iqi\Delta x} + \xi^n e^{Iqi\Delta x}\end{aligned}$$

Dividing through by $\xi^n e^{Iqi\Delta x}$ we get

$$\xi[1 + C\frac{\rho\Delta t}{3}] = R(e^{Iq\Delta x} - 2 + e^{-Iq\Delta x}) + \rho\Delta t + 1$$

$$\implies \xi[1 + C\frac{\rho\Delta t}{3}] = 2R(\frac{e^{Iq\Delta x} + e^{-Iq\Delta x}}{2} - 1) + \rho\Delta t + 1$$

Now using Euler's formula and the double angle formula for $\cos(\theta)$ we get,

$$\xi[1 + C\frac{\rho\Delta t}{3}] = -4R(\sin^2(\frac{q\Delta x}{2})) + \rho\Delta t + 1$$

$$\implies \xi = \frac{1 - 4R(\sin^2(\frac{q\Delta x}{2})) + \rho\Delta t}{[1 + C\frac{\rho\Delta t}{3}]}$$

For our scheme to be stable, we need the amplification factor $-1 < \xi < 1$. The condition we need for Δx and Δt such that $-1 < \xi < 1$ holds is,

$$-1 < \frac{1 - 4R(\sin^2(\frac{q\Delta x}{2})) + \rho\Delta t}{[1 + C\frac{\rho\Delta t}{3}]} < 1 \tag{10}$$

Now taking the RHS of equation (10) we have that

$$1 - 4R(\sin^2(\frac{q\Delta x}{2})) + \rho\Delta t < 1 + C\frac{\rho\Delta t}{3}$$

We note that $\max(\sin^2(\frac{q\Delta x}{2})) = 1$. Thus

$$-4R(\sin^2(\frac{q\Delta x}{2})) < C\frac{\rho\Delta t}{3} - \rho\Delta t$$

We choose $C = 0$, then

$$\rho\Delta t < 4R$$

$$\implies R > \frac{\rho\Delta t}{4}$$

Now consider the LHS of equation (10)

$$-1 < \frac{1 - 4R(\sin^2(\frac{q\Delta x}{2})) + \rho\Delta t}{[1 + C\frac{\rho\Delta t}{3}]}$$

Since $max(\sin^2(\frac{q\Delta x}{2})) = 1$ and $C = 0$, we obtain

$$4R < 2 + \rho\Delta t$$

$$\implies R < \frac{2 + \rho\Delta t}{4}$$

Thus the Non-Standard Finite Difference scheme is conditionally stable for $\frac{\rho\Delta t}{4} < R < \frac{2+\rho\Delta t}{4}$. Expressed in terms of Δx and Δt

$$\frac{4\Delta t}{2 + \rho\Delta t} < \Delta x^2 < \frac{4}{\rho} \tag{11}$$

We note that this condition is identical to that imposed by the Finite Difference scheme. □

3.2.3 Convergence

. By the Lax Equivalence Theorem, the scheme is convergent. □

4 Results

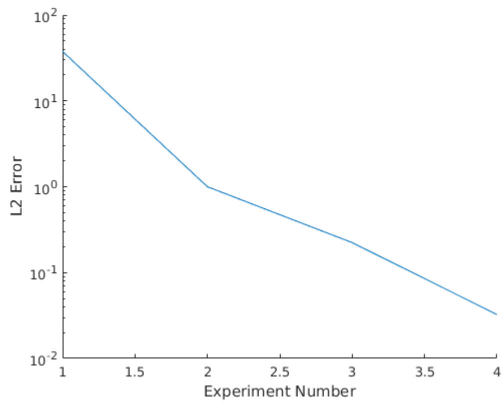
4.1 Finite Difference Method

Comparing the Finite Difference scheme to the true solution, we obtained the following errors at $t = 0.005$, using the initial condition of $u(x, 0) = \frac{1}{(\exp 10\sqrt{\frac{10}{3}}x+1)^2}$ and Dirichlet boundary conditions $u(-1, t) = 1$ and $u(1, t) = 0$, with $\rho = 2000$, we get the following:

Experiment	Δx	Δt	L_2 Error
1	0.05	0.00125	37.6239
2	0.025	0.0003125	1.0009
3	0.0125	0.000078125	0.2235
4	0.00625	0.0000195313	0.0324

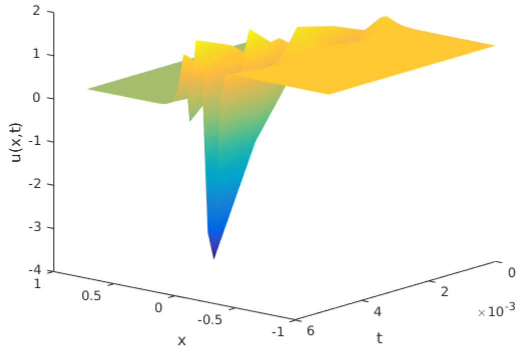
Plotting the errors against the experiment number, we received the following results.

Figure 2: L_2 Error vs. Experiment Number



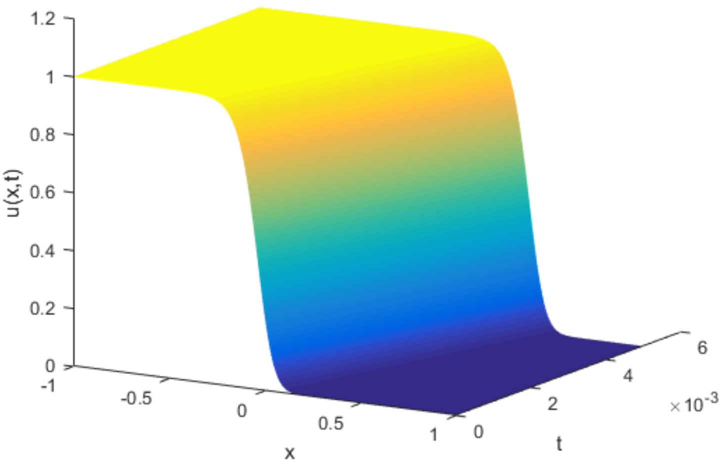
The huge error for experiment one is indicative of the numerical instability caused by the overly large step sizes. This instability is depicted in the mesh plot below.

Figure 3: Numerical Instability in the Finite Difference Approach



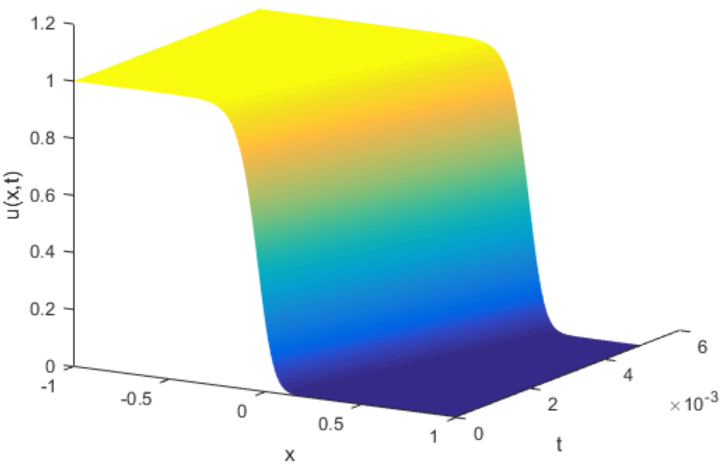
This instability is expected, as the value for Δx in experiment 1 does not satisfy the stability condition outlined in (11).
Plotting a mesh plot of the scheme in the Dirichlet Boundary case, using the values $\Delta x = 0.00625$ and $\Delta t = 0.0000195313$ produces the following.

Figure 4: Finite Difference Approximation with Dirichlet Boundaries



The Neumann Boundary case, $u_x(-1,t) = 0$ and $u_x(1,t) = 0$, for the same values of Δx and Δt the plot as follows:

Figure 5: Finite Difference Approximation with Neumann Boundaries



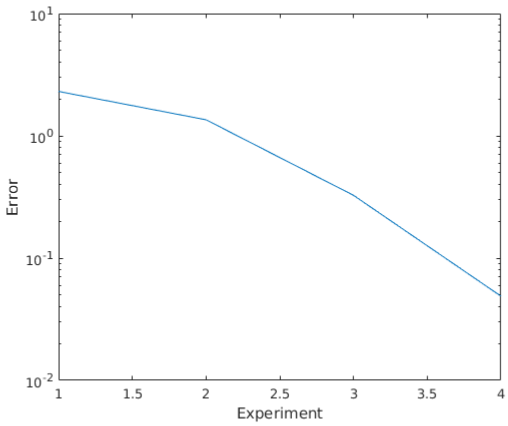
4.2 Non-Standard Finite Difference Method

Comparing the Non-Standard Finite Difference scheme to the true solution, we obtained the following errors at $t = 0.005$, using the initial condition of $u(x,0) = \frac{1}{(\exp 10\sqrt{\frac{10}{3}}x+1)^2}$ and Dirichlet boundary conditions $u(-1,t) = 1$ and $u(1,t) = 0$, with $\rho = 2000$, we get the following:

Experiment	Δx	Δt	L_2 Error
1	0.05	0.00125	2.3075
2	0.025	0.0003125	1.3537
3	0.0125	0.000078125	0.3262
4	0.00625	0.0000195313	0.0488

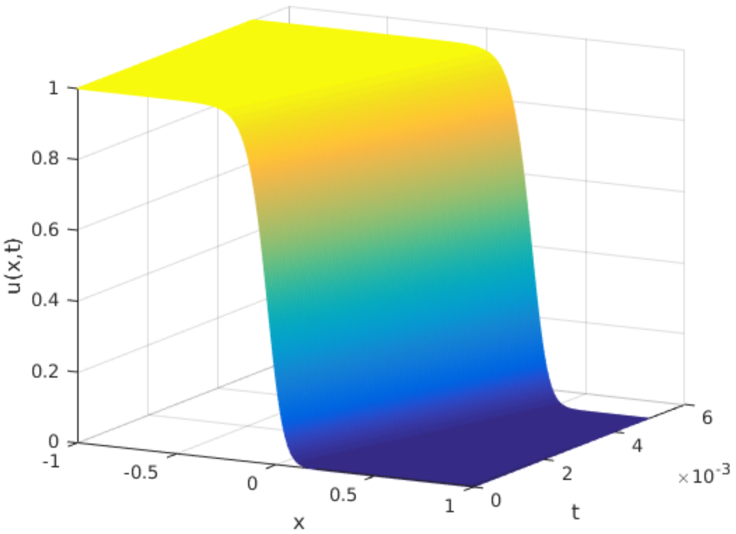
Plotting the errors against the experiment number, we received the following results.

Figure 6: L_2 Error vs. Experiment Number



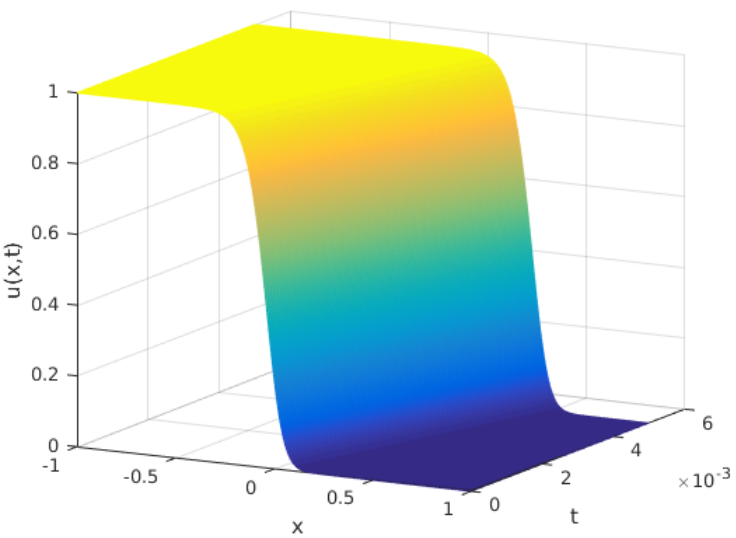
Plotting a mesh plot of the scheme in the Dirichlet Boundary case, using the values $\Delta x = 0.00625$ and $\Delta t = 0.0000195313$ produces the following.

Figure 7: Non-Standard Finite Difference Approximation with Dirichlet Boundaries



The Neumann Boundary case, $u_x(-1,t) = 0$ and $u_x(1,t) = 0$, for the same values of Δx and Δt the plot as follows:

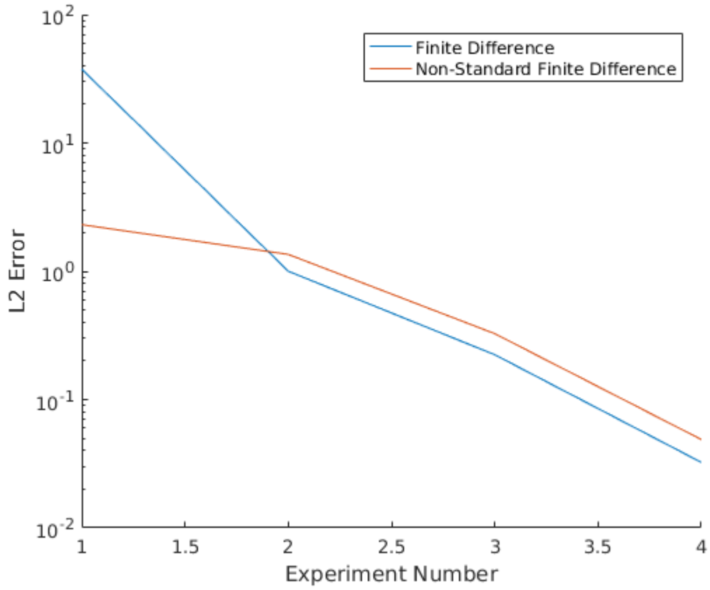
Figure 8: Non-Standard Finite Difference Approximation with Neumann Boundaries



5 Discussion

Both methods approximate the solution, a travelling wave front, quite well, numerical instabilities aside.
Plotting the errors from section 4.1 and 4.2 on a single graph, we obtain the following:

Figure 9: L_2 Error vs. Experiment Number



Clearly, the Non-Standard Finite Difference approach vastly outperforms the Finite Difference approach in the first experiment, where the step sizes Δx and Δt are relatively large. However, from experiment 2 onward, where the step sizes are sufficiently small to ensure stability for both methods during the time interval, the Finite Difference approach approximates the solution slightly better than the Non-Standard Finite Difference approach. The fact that the two methods perform so similarly is not surprising, given that they differ only in their expression for u^2 and that they have identical conditions for stability, given our linearisations.
In terms of time complexity, both algorithms are $\mathcal{O}(TN)$, where N is the number of spatial points and T is the number of temporal points. Thus neither algorithm can be said to be better or worse with regards to computation.

6 Conclusion

Having analysed at length both the Finite Difference and Non-Standard Finite Difference approaches using both theoretical and experimental techniques, we conclude that, while both approaches are very similar, the Non-Standard Finite Difference method is more robust, coping better with overly large step sizes while performing admirably with small ones, at least in the case of the Fisher Equation.

Although the Finite Difference method performs better than the Non-Standard Finite Difference method in this case, the fact that it succumbs to numerical instability more easily does not recommend it against the Non-Standard scheme. Furthermore, its performance hinges on small step sizes, therefore requiring many more points for the same interval. Depending on the size of the space and time intervals, such small step sizes may not be feasible.

Since the two methods perform so similarly, we suggest further research into their comparison. In particular, we suggest performing the experimental analysis for many more configurations of Δx and Δt, in order to truly gauge their effect on the methods' efficacy. We further suggest that the robustness of each method be investigated over longer time intervals, where the effects of minor errors and instabilities become more pronounced. Finally, we suggest extending these methods to other equations, both within the Reaction-Diffusion class and beyond, in order to determine their performance across multiple equations.

References

[1] Mickens, Ronald E. Nonstandard finite difference schemes for reaction-diffusion equations. *Numerical Methods for Partial Differential Equations: An International Journal* 15.2 (1999): 201-214.

[2] Mickens, Ronald E. Applications of nonstandard finite difference schemes. *World Scientific*, 2000.

[3] Anguelov, R., P. Kama, and JM-S. Lubuma. On non-standard finite difference models of reaction–diffusion equations. *Journal of computational and Applied Mathematics* 175.1 (2005): 11-29.

[4] Fisher, R. The Wave of Advance of Advantageous Genes. *Annals of Human Genetics*, June 1937.

Appendices

A Finite Difference Method Code

A.1 Dirichlet Case

```
%Fisher Equation - Dirichlet
clc
close all

p = 2000;
f = @(x) 1./((exp(10*sqrt(10/3).*x)+1).^2);
sol = @(x,t) 1./((exp(sqrt(p/6).*x-(5*p/6)*t)+1).^2);

error = zeros(1,4);

for ex = 1:4

    switch ex
    case 1
        dx = 0.05;
        dt = 0.00125;
    case 2
        dx = 0.025;
        dt = 0.0003125;
    case 3
        dx = 0.0125;
        dt = 0.000078125;
    case 4
        dx = 0.00625;
        dt = 0.0000195313;
    end

    xs = -1:dx:1;
    N = length(xs);
    time = 0:dt:0.005;
    T = length(time);
    R = (dt)/(dx^2);

    u = f(xs);
    u(1) = 1;
    u(end) = 0;

    A = zeros(T,N) ;
    %B = zeros(T,N) ;

    for t = 1:T
        s = sol(xs,time(t));
        old = u;
        u(2:end-1) = old(2:end-1) + R*(old(3:end)-2*old(2:end-1)+
    old(1:end-2));
        u(2:end-1) = u(2:end-1) + p*dt*old(2:end-1).*(ones(1,N-2)-
    old(2:end-1));

        A(t,:) = u;
        %B(t,:) = s;
        %Plot
%           figure(1);
%           plot(xs,u,xs,s);
%           axis([-1,1,0,1])
```

```
%           pause(0.01);
    end

    error(ex) = sum((u - s).^2);
end

%hold off

%   Mesh Plot
[xs,time] = meshgrid(xs,time) ;
hold on
surf(xs,time,A);
%surf(xs,time,B);
%title('Finite Difference Approximation of u(x,t) vs. x vs. t - Dirichlet Boundaries');
xlabel('x');
ylabel('t');
zlabel('u(x,t)');
hold off;
shading interp

%   Error Plot
% hold on
% plot(1:4,error);
% set(gca,'yscale','log')
% title('Error Plot for Finite Difference Approach');
% xlabel('Experiment Number');
% ylabel('L2 Error');
% hold off
```

Fisher_Dirichlet_FiniteDifference.m

A.2 Neumann Case

```
%Fisher Equation - Neumann
clc
close all

p = 2000;
f = @(x) 1./((exp(10*sqrt(10/3).*x)+1).^2);
sol = @(x,t) 1./((exp(sqrt(p/6).*x-(5*p/6)*t)+1).^2);

ex = 4;

switch ex
    case 1
        dx = 0.05;
        dt = 0.00125;
    case 2
        dx = 0.025;
        dt = 0.0003125;
    case 3
        dx = 0.0125;
        dt = 0.000078125;
    case 4
        dx = 0.00625;
        dt = 0.0000195313;
end

xs = -1:dx:1;
N = length(xs);
```

```
time = 0:dt:0.005;
T = length(time);
R = (dt)/(dx^2);

u = zeros(1,length(xs)+2);
u(2:end-1) = f(xs);
u(1) = u(2);
u(end) = u(end-1);

A = zeros(T,N) ;
%B = zeros(T,N) ;

for t = 1:T
    s = sol(xs,time(t));
    old = u;
    u(1) = u(3);
    u(end) = u(end-2);
    u(2:end-1) = old(2:end-1) + R*(old(3:end)-2*old(2:end-1)+old(1:
    end-2));
    u(2:end-1) = u(2:end-1) + p*dt*old(2:end-1).*(ones(1,N)-old(2:
    end-1));

    A(t,:) = u(2:end-1);
    %B(t,:) = s;
    %Plot
%     figure(1);
%     plot(xs,u(2:end-1),xs,s);
%     axis([-1,1,0,1])
%     pause(0.001);
end

%hold off

%   Mesh Plot
[xs,time] = meshgrid(xs,time) ;
hold on
surf(xs,time,A);
%surf(xs,time,B);
%title('Finite Difference Approximation of u(x,t) vs. x vs. t -
    Neumann Boundaries');
xlabel('x');
ylabel('t');
zlabel('u(x,t)');
hold off;
shading interp
```

Fisher_Neumann_FiniteDifference.m

B Non-Standard Finite Difference Method Code

```
experiment = 4;
err = zeros(1, experiment);

bc = 'D';
%bc = 'N';
isSimulate = false;
isPlot3D = false;


for i = 1:experiment
  fprintf('Experiment: %i\n', i)
  switch i
    case 1
          dx = 0.05;
          dt = 0.00125;
    case 2
          dx = 0.025;
          dt = 0.0003125;
    case 3
          dx = 0.0125;
          dt = 0.000078125;
    case 4
          dx = 0.00625;
          dt = 0.0000195313;
  end

  if bc == 'D'
    [curr_error] = NSFD_Dirichlet(dx, dt, i, isSimulate, isPlot3D
  );
  else
    [curr_error] = NSFD_Neumann(dx, dt, i, isSimulate, isPlot3D)
  end
  err(i) = curr_error

end

% Plot Error
figure(3)
plot([1:4], err)
set(gca, 'yscale', 'log')
xlabel('Experiment')
ylabel('Error')
```

main_NSFD.m

B.1 Dirichlet Case

```
function [err] = NSFD_Dirichlet(dx, dt, experiment, isSimulate,
    isPlot3D)

  p = 2000;
  f = @(x) 1./((exp(10*sqrt(10/3)*x)+1).^2);
  sol = @(x,t) 1./((exp(sqrt(p/6).*x-(5*p/6)*t)+1).^2);

  xs = -1:dx:1;
  time = 0:dt:0.005;
  T = length(time);
  p = 2000;
```

```
  R = (dt)/(dx^2);

  un = f(xs);
  un(1) = 1;
  un(end) = 0;
  u_points = zeros(length(T), length(xs));

  error_exp = zeros(1, T);
  err = 0;


  for n = 1:T

    old = un;
    C = (1+p*dt-2*R);
    un(2:end-1) = (R.*(old(3:end)+old(1:end-2)) + C.*old(2:end-1))
    ./(ones(1,length(xs)-2)+((p*dt)/3).*(old(3:end)+old(2:end-1)+
    old(1:end-2)));
    u_points(n, :) = un;

    % true solution
    u_sol = sol(xs,time(n));

    error_exp(n) = sum((un-u_sol).^2);

    if(n == T)
      err = error_exp(n);
    end

    % simulation
    if isSimulate
      figure(1)
      plot(xs, un, xs, u_sol);
      legend({'NSFD solution','True solution'},'Location','
    northeast')
      title(['Experiment ', num2str(experiment)])
      axis([-1 1 0 1])
      pause(0.01);
    end

  end

  % 3D plot
  if isPlot3D
    figure(2)
    mesh(xs, time, u_points)
    xlabel('x')
    ylabel('t')
    zlabel('u(x,t)')
  end

% figure(4)
% plot([1:T], error_exp)

end
```

NSFD_Dirichlet.m

B.2 Neumann Case

```
function [err] = NSFD_Neumann(dx, dt, experiment, isSimulate,
    isPlot3D)

  p = 2000;
  f = @(x) 1./((exp(10*sqrt(10/3)*x)+1).^2);
  sol = @(x,t) 1./((exp(sqrt(p/6).*x-(5*p/6)*t)+1).^2);

  xs = -1:dx:1;
  time = 0:dt:0.005;
  T = length(time);
  p = 2000;
  R = (dt)/(dx^2);

  un = zeros (1, length(xs)+2);
  un(2:end-1) = f(xs);
  un(1) = un(2);
  un(end) = un(end-1);

  error_exp = zeros(1, T);
  err = 0;

  u_points = zeros(length(time), length(xs));
  for n = 1:T

    old = un;
    un(1) = un(3);
    un(end) = un(end-2);

    C = (1+p*dt-2*R);
    un(2:end-1) = (R.*(old(3:end)+old(1:end-2)) + C.*old(2:end-1))
    ./(ones(1,length(xs))+((p*dt)/3).*(old(3:end)+old(2:end-1)+old
    (1:end-2)));
    u_points(n, :) = un(2:end-1);

    % true solution
    u_sol = sol(xs,time(n));

    error_exp(n) = sum((un(2:end-1)-u_sol).^2);

    if(n == T)
      err = error_exp(n);
    end

    % simulation
    if isSimulate
      figure(1)
      plot(xs, un(2:end-1), xs, u_sol);
      legend({'NSFD solution','True solution'},'Location','
    northeast')
      title(['Experiment ', num2str(experiment)])
      axis([-1 1 0 1])
      pause(0.01);
    end

  end

   if isPlot3D
     figure(2)
     mesh(xs, time, u_points)
     xlabel('x')
     ylabel('t')
     zlabel('u(x,t)')
```

```
    end

%   figure(4)
%   plot([1:T], error_exp)


end
```

NSFD_Neumann.m