Blow-up in a Chemotaxis Model Using a Moving Mesh Method

UNIVERSITY OF READING
Department of Mathematics

Sarah Lianne Cole

Supervisor: Professor Michael J. Baines

Abstract

In this dissertation we look at a system of partial differential equations (PDEs) used to model chemotaxis. This model is called the Keller-Segel model. The solution to this model exhibits interesting properties. In particular the solution becomes infinite in a finite time, T. Firstly we look at a simpler equation, the Fishers equation, to see if blow-up exists. The aim of this project is to look at numerical methods that will capture the solution as it blows up. We see in the paper by Budd et al, [1] how a moving mesh PDE (MMPDE) is used to track the solution as it blows up. We then look at a different moving mesh method, which conserves the relative area underneath the solution curve. This method is applied to the Fishers equation, to demonstrate how it works. The method is then applied to the Keller-Segel model. The results obtained are then discussed.

Acknowledgements

Firstly, I would like to thank my supervisor Mike Baines for his endless ideas, support and enthusiasm, he is an incredible asset to the Maths department, and it has been a pleasure to work with him. Also, thank you to my co-supervisor Dr Marcus Tindall for his additional advice.

I must thank all the wonderful people I have met at Reading University for giving me the best year of my life, so far. Special thanks go to all the Maths postgraduates in particular my fellow MSc students for all the laughs shared in the computer room, and the countless cups of tea.

To mio piccolo fiorellino, Dani, who is never too busy to listen. You have been my ray of sunshine this year. Thank you for the late night chats, supporting me through the highs and lows and encouraging me to further my studies. Thank you for being you.

Thank you to my loving family and friends for always believing in me, encouraging me to go further and being there when I've needed you.

Finally I would like to acknowledge the EPSRC for their financial support for this MSc course.

Declaration

I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Signed...Date...

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 What is Chemotaxis?

Chemotaxis describes the movement of single or multicellular organisms when they move up or down a chemical gradient. The suffix ‘taxis’ is Greek for ‘arrange, turning’, [1]. This movement allows the organism to explore its extracellular environment. Organisms move randomly, away from repellents and towards attractants. Questions have arisen on how organisms can detect small changes in their extracellular environment. Usually the organism will undergo a random walk, consisting of smooth swimming and brief direction changes (tumbles). By increasing the attractant, the tumbling is suppressed, which leads to a biased random walk. The organism will then accumulate in areas of high attractant concentration. This type of movement is referred to as runs. A combination of tumbles and runs allows the organism to explore and respond to changes in its extracellular environment, as explained in [9]

There is much interest in the study of chemotaxis for many reasons. Chemotaxis is important in multicellular organisms, as it is critical in certain phases of development e.g. during fertilisation, as the sperm moves towards the egg, from [4] and explained further in [10]. Chemotaxis may also underpin the pigmentation patterning in snakes and fish. Whilst modelling certain phases of tumour growth, chemotaxis has been incorporated into these models, explained further in [10].

1.2 Keller-Segel Model

The modelling of chemotaxis "has developed into a large and diverse discipline" [4]. One model which is widely used is the "Keller-Segel model of chemotaxis" see [4] and explained further in [12] . A property of this model is that it can display 'auto-aggregation'[4], which leads to a blow-up solution in a finite time T, which is what we wish to investigate in this project.

Originally in the 1970s the Keller-Segel model consisted of four coupled reaction-advection-diffusion equations. After reduction under quasi-steady-state assumptions we have a model for two unknown functions u and v, see [10].

The general form of the model is

$$\begin{aligned} u_t &= \nabla(k_1(u,v)\nabla u - k_2(u,v)u\nabla v) + k_3(u,v), \\ v_t &= D_v\Delta v + k_4(u,v) - k_5(u,v)v \end{aligned}$$

where

- u denotes the cell density on a given domain $\Omega \subset \mathbb{R}^n$
- v describes the concentration of the chemical substrate
- $k_1(u,v)$ describes the diffusivity of the cells
- $k_2(u,v)$ is the chemotactic sensitivity
- $k_3(u,v)$ describes cell growth and death

- $k_4(u, v)$ and $k_5(u, v)$ describes the production and degradation of the chemical substrate

- D is the diffusion coefficient of the attractant

There has been a large amount of work on the conditions which form either finite-time blow-up or have globally existing solutions. This has resulted in the model which is referred to as the *'minimal model'*, where it is assumed that the functions k_j have linear form (see [4], for further information). We shall be concerned with this minimal model due to its finite-time blow-up property.

The minimal model (which is non-dimensional) is

$$\begin{aligned} u_t &= \nabla^2 u - \chi \nabla.(u \nabla v) \\ v_t &= \nabla^2 v + u - v \end{aligned} \tag{1.1}$$

where

- $\mathbf{r} \in \Omega = \{\mathbf{r} : |\mathbf{r}| \leq R\}$, the region is suggested to be [0,1]

- $u(\mathbf{r}, t)$ is the evolution of cell density

- $v(\mathbf{r}, t)$ is the chemical substrate

- χ chemotactic coefficient, whose suggested value is $\chi = 8$

- Neumann boundary conditions are enforced for u and v on $\partial\Omega$

see[1].

We restrict our attention to radially symmetric versions of (1.1).

1.3 Initial Conditions

Following Budd et al [1], we take Gaussian functions for our initial data. They will be of the form

$$u(r,0) = 1000e^{-500r^2}$$

$$v(r,0) = 10e^{-500r^2}$$

over the domain [0,1], with data taken from [1].

Numerical models have been developed so that model predictions can be compared with experimental data. We wish to develop a numerical method to successfully capture the blow-up of the solution to the Keller-Segel model.

We begin by studying the existence of blow-up for a simpler equation, the nonlinear Fisher's equation in the next two chapters, before using the same technique to study Chemotaxis.

Chapter 2

Blow-up and the Fisher equation

We begin our study on blow-up in chemotaxis by considering the Fisher's equation, on a fixed 1D cartesian mesh.

2.1 Fisher's Equation

Fisher's equation is a diffusion equation with an added source term, i.e.

$$u_t = u_{xx} + u^p, \qquad (p > 1) \tag{2.1}$$

Equation (2.1) represents the temperature of a reacting or combusting medium. This equation is of particular interest since its solution is simpler than the Keller-Segel model and also becomes infnite in a finite time, i.e. blow-up exists. Usually a problem that exhibits blow-up will become infinite at a single blow-up point. The blow-up point x^*, is described in [2] and occurs at a finite blow up time, T, where $T < \infty$, so as $t \to T$

$$u(x^*, t) \to \infty \qquad \text{and} \qquad u(x, t) \to u(x, T) < \infty, \qquad \text{if} \qquad x \neq x^*$$

see [2]

2.2 Model Formation

As in [2] we shall consider the model problem (2.1) where $p = 2$, i.e.

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

in $x \in [0, 1]$, with boundary conditions

$$u(0, t) = u(1, t) = 0, \qquad u(x, 0) = u_0(x) > 0$$

We take an initial condition which according to [2] is such that blow-up occurs, with a single blow-up point at $x^* = 0.5$, which is where the maximum of the function occurs,

$$u(x, 0) = 20sin(\pi x)$$

where $x^* = 0.5$.

2.3 Explicit Method

To study the blow-up behaviour of Fisher's equation we will firstly approximate the solution on a *fixed* mesh, using a standard explicit and implicit scheme. We first look at an explicit method on a fixed mesh, to obtain a solution.

By using the method of finite differences we can discretise (2.1) in space and time to obtain;

$$\begin{aligned} \frac{(u_j^{n+1} - u_j^n)}{\Delta t} &= \frac{(u_{j+1}^n - 2u_j^n + u_{j-1}^n)}{\Delta x^2} + (u_j^n)^2 \\ \Rightarrow u_j^{n+1} &= u_j^n + \frac{\Delta t}{\Delta x^2}\left[u_{j+1}^n - 2u_j^n + u_{j-1}^n\right] + \Delta t(u_j^n)^2 \end{aligned}$$

Note that $\frac{\Delta t}{\Delta x^2}$ is restricted for reasons of numerical stability. From this we obtain the solution shown in Figure (2.1).

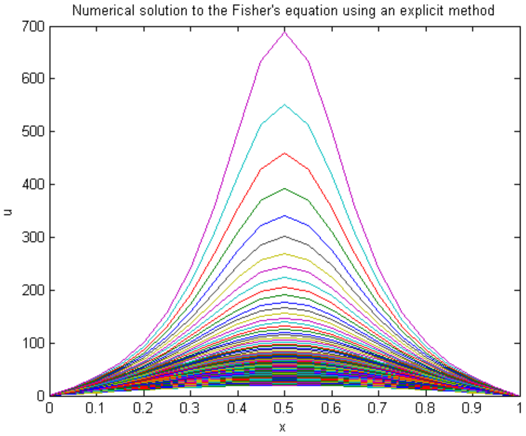


Figure 2.1: Solution to the Fisher's equation using an explicit method, where $\Delta t = 0.0005$, $nt = 165$ and $\Delta x = 0.05$

Although the solution is not very refined due to the small number of nodes along the x-axis, we can begin to see the blow-up at $x^* = 0.5$. With the explicit method we need to take a small Δt and many timesteps, to see the solution blow-up.

Clearly, as blow-up continues the fixed mesh will not be able to resolve the peak, since the width of the peak may become less than the distance between the nodes.

2.4 Implicit Method

Since Δt is rather restricted in Section (2.3) we now use an implicit method on a fixed mesh to solve the equation. This is done by discretising the second order derivative u_{xx} at the forward time. This leads to

$$\begin{aligned} \frac{(u_j^{n+1} - u_j^n)}{\Delta t} &= \frac{(u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1})}{\Delta x^2} + (u_j^n)^2 \\ &\Rightarrow \quad u_j^{n+1} - \frac{\Delta t}{\Delta x^2}\left[u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}\right] = u_j^n + \Delta t (u_j^n)^2 \end{aligned}$$

which leads to a tridiagonal matrix which can be solved for u_j^{n+1}

$$A\underline{u}_j^{n+1} = \underline{u}_j^n + \Delta t(\underline{u}_j^n)^2$$

where

$$A = \begin{pmatrix} 1+\frac{2\Delta t}{\Delta x^2} & -\frac{\Delta t}{\Delta x^2} & 0 & \cdots & \cdots & 0 \\ -\frac{\Delta t}{\Delta x^2} & 1+\frac{2\Delta t}{\Delta x^2} & -\frac{\Delta t}{\Delta x^2} & 0 & \cdots & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \ddots & \ddots & -\frac{\Delta t}{\Delta x^2} & 1+\frac{2\Delta t}{\Delta x^2} & -\frac{\Delta t}{\Delta x^2} & \ddots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \ddots & \ddots & \ddots & -\frac{\Delta t}{\Delta x^2} & 1+\frac{2\Delta t}{\Delta x^2} \end{pmatrix}$$

This method is more stable. We obtain the solution shown in Figure (2.2), with larger values of Δt.

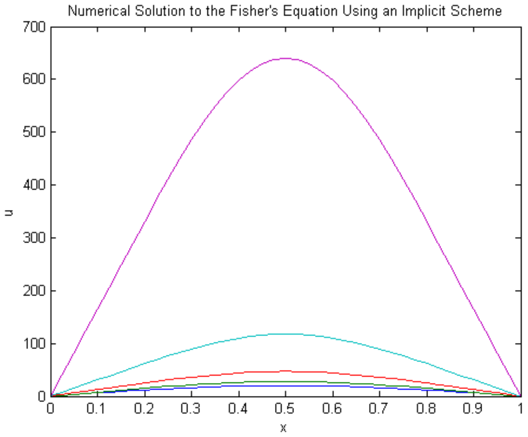


Figure 2.2: Solution to the Fisher's equation using an implicit method, where $\Delta t = 0.1$, $nt = 5$ and $\Delta x = 0.025$

Once again we can observe the solution blow-up at the point $x^* = 0.5$. Since the implicit method is more stable than the explicit one, we can take bigger Δt and fewer timesteps. This makes it less computationally expensive. Even so, the peak will eventually not be resolved with a fixed mesh.

Many numerical methods have been developed to refine a solution as it blows up, in particular moving mesh methods. We will now apply a moving mesh method to the Fisher's equation, using a method based on conservation.

Chapter 3

Method of Conservation for Fisher's Equation

Previously in Sections (2.3) and (2.4) we have seen how a fixed mesh can be used to investigate the solution to (2.1). From Figure (2.1) and Figure (2.2), as $t \to T$ we can see a single spike forming around $x^* = 0.5$. The spike is increasing in height and decreasing in width. The width of the spike will become smaller and smaller as $t \to T$. This means that on a fixed mesh the width of the spike will eventually be smaller than the size of the mesh. In this case the numerical method will not be able to resolve the blow-up, which is why it is essential to use an adaptive method. In this way nodes can be moved in towards the spike, which can resolve the blow-up solution more accurately.

For blow-up problems it is natural to use an adaptive mesh procedure, where mesh points are clustered close to to the region where blow-up occurs, to be able to resolve the solution. Various adaptive procedures have previously been used, including:

(1) h-refinement- this is on a static mesh. The mesh is refined by adding

nodes to the area where the singularity occurs. This becomes computationally expensive as the blow-up develops, and its width decreases.

(2) p-refinement- this is also static. It uses higher order polynomials to get an accurate representation of the solution. It is more accurate in each cell than h-refinement. However, a polynomial will not be able to model blow-up if it falls between nodes.

(3) r-refinement- this is a moving mesh method where a fixed number of nodes are moved into the region where the singularity occurs. The advantage of this type of refinement is that it can keep track of the singularity all the way up to blow-up time. This refinement is not expensive to compute, but the solution away from the blow-up can be poorly tracked as there are fewer nodes in these regions.

To track the solution as it blows up, we will use a moving mesh, aiming to conserve the relative or fractional area under the curve, as time goes on.

We will use a velocity based method to move the individual nodes at each timestep, which will track the solution as it blows up. We obtain the velocities by conserving the mass of each element under the solution curve, for each timestep.

3.1 Generating Velocities

By dividing the x domain $[0,1]$ into N regions $(x_{j-1}(t), x_j(t)), j = 1, 2, .., N$, we can obtain the areas in these regions underneath the solution curve by

$$\int_{x_{j-1}(t)}^{x_j(t)} u(x,t)dx = \text{area}_j \tag{3.1}$$

By adding the individual areas we can obtain the area under the entire solution curve, which we call θ, i.e.

$$\theta(t) = \int_0^1 u(x,t)dx \tag{3.2}$$

By differentiating (3.2) with respect to time, the rate of change of θ is given by

$$\dot{\theta} = \int_0^1 u_t dx$$

We will use Fishers equation, (2.1), to substitute for u_t, giving

$$\begin{aligned} \dot{\theta} &= \int_0^1 (u_{xx} + u^2)dx \qquad (3.3) \\ &= [u_x]_0^1 + \int_0^1 u^2 dx \end{aligned}$$

In the method of conservation the relative or fractional area

$$\frac{1}{\theta}\int_{x_{j-1}(t)}^{x_j(t)} u(x,t)dx \tag{3.4}$$

of the regions under the solution curve throughout the evolution is held constant, therefore by differentiating (3.4) with respect to time

$$\frac{d}{dt}\left[\frac{1}{\theta(t)}\int_{x_{j-1}(t)}^{x_j(t)} u(t)dx\right] = 0 \tag{3.5}$$

Since we are differentiating under the integral sign we use *Leibniz Integral Rule* and obtain

$$0 \;=\; -\frac{1}{\theta^2}\dot{\theta}\int_{x_{j-1}(t)}^{x_j(t)} u(t,x)dx + \frac{1}{\theta}\left[\int_{x_{j-1}(t)}^{x_j(t)} \frac{\partial u}{\partial t}dx + [uv]_{x_{j-1}}^{x_j}\right]$$

$$\begin{aligned}
&= -\frac{\dot{\theta}}{\theta}\int_{x_{j-1}(t)}^{x_j(t)} u(t,x)dx + \int_{x_{j-1}(t)}^{x_j(t)} (u_{xx}+u^2)dx + [uv]_{x_{j-1}}^{x_j} \\
&= -\frac{\dot{\theta}}{\theta}\text{area}_j + [u_x]_{x_{j-1}}^{x_j} + \int_{x_{j-1}(t)}^{x_j(t)} u^2 dx + [uv]_{x_{j-1}}^{x_j} \\
&= -\frac{\dot{\theta}}{\theta}\text{area}_j + [u_x]_{x_{j-1}}^{x_j} + \int_{x_{j-1}(t)}^{x_j(t)} u^2 dx + u_j v_j - u_{j-1}v_{j-1}
\end{aligned}$$

Where area_j is given by (3.1), this equation can then be solved sequentially for the velocity v_j at a certain node j. Since $v_N = 0$ and everything else is known. This gives the recursion formula

$$v_j = -\frac{1}{u_j}\left[-\frac{\dot{\theta}}{\theta}\text{area}_j + [u_x]_{x_{j-1}}^{x_j} + \int_{x_{j-1}(t)}^{x_j(t)} u^2 dx - u_{j-1}v_{j-1}\right] \tag{3.6}$$

The integral in (3.1) and (3.6) can be approximated by the trapezium rule.

3.2 Recovering new values of x and θ

Noting that

$$v = \frac{dx}{dt} \tag{3.7}$$

By using explicit Euler time stepping we can obtain new x and θ values,

$$x_j^{n+1} = x_j^n + \Delta t v_j^n$$

$$\theta_j^{n+1} = \theta_j^n + \Delta t \dot{\theta}_j^n$$

3.3 Recovering new values of u

Since from (3.4) we can deduce that

$$\frac{1}{\theta(t)}\int_{x_{j-1}(t)}^{x_{j+1}(t)} u(t)dx$$

is constant in time, equal to its initial value, we can recover new values for u. By combining adjacent integrals we have

$$\frac{1}{\theta(t)}\int_{x_{j-1}(t)}^{x_{j+1}(t)} u(t)dx \quad = \quad \frac{1}{\theta(0)}\int_{x_{j-1}(0)}^{x_{j+1}(0)} u(0)dx$$

which can be approximated by the mid-point rule to give

$$\begin{aligned} \frac{1}{\theta(t)}u_j(t)[x_{j+1}(t) - x_{j-1}(t)] &= \frac{1}{\theta(0)}u_j(0)[x_{j+1}(0) - x_{j-1}(0)] \\ \Rightarrow u_j(t) &= \frac{\theta(t)[x_{j+1}(0) - x_{j-1}(0)]}{\theta(0)[x_{j+1}(t) - x_{j-1}(t)]} \end{aligned}$$

Chapter 4

Results for Fisher's Equation

To investigate the behaviour of the Fisher's equation using a moving mesh method, we look at various combinations of Δt_0 and Δx. We use a variable timestep for reasons of numerical stability. We will have Δt varying as $t \to T$. We take,

$$\Delta t = \frac{\Delta t_0}{T - t}$$

however, the downfall with this is that we need an estimation of T. In this case we will use $T \approx 0.082372$ from [2].

Firstly we fix Δx and choose a Δt_0, which returns stable results. We then approach the blow-up time by increasing the number of timesteps, nt. After this we almost halve Δt_0, and again approach the blow-up time. The whole process is then repeated for a smaller Δx. The results obtained are as follows:

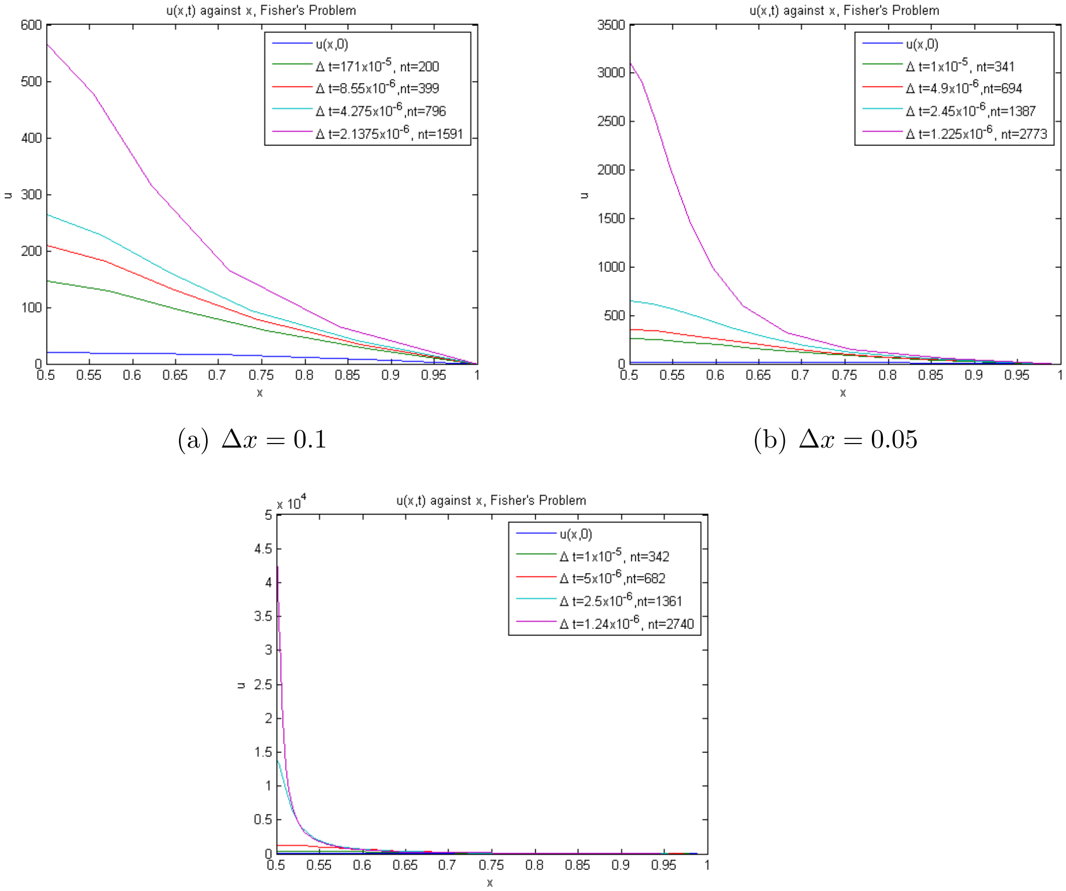
Table 4.1: Changing Δt_0, nr and nt for the Fisher's Equation

nx	Δt_0	nt	T
11	1.71×10^{-5}	200	0.0774
11	8.55×10^{-6}	399	0.0797
11	4.275×10^{-6}	796	0.0806
11	2.1375×10^{-6}	1591	0.0836
21	1×10^{-5}	341	0.0786
21	4.9×10^{-6}	694	0.0793
21	2.45×10^{-6}	1387	0.0807
21	1.225×10^{-6}	2773	0.0824
41	1×10^{-5}	342	0.0812
41	5×10^{-6}	682	0.0835
41	2.5×10^{-6}	1361	0.0847
41	1.24×10^{-6}	2740	0.0836

From (4.1) we can see that when $nx = 11$ and we decrease Δt_0 and increase the number of timesteps nt, we are approaching the blow-up time, which is almost in agreement with Budd et al [1], where $T \approx 0.082372$.

By incresing and keeping Δt_0 fixed, e.g. when $= 21$ and $nx = 41$ and with $\Delta t_0 = 10^{-5}$ we see T is increasing, and again is close to $T \approx 0.082372$.

We plot $u(x, t)$ at the final time for each combination of Δt_0 and nt. Graphically the results obtained are as follows: As Δt_0 is decreased and nt increased, i.e.Figure (4.1), we see the solution convreging. By decreasing Δx the solution $u(x, t)$ converges more rapidly.



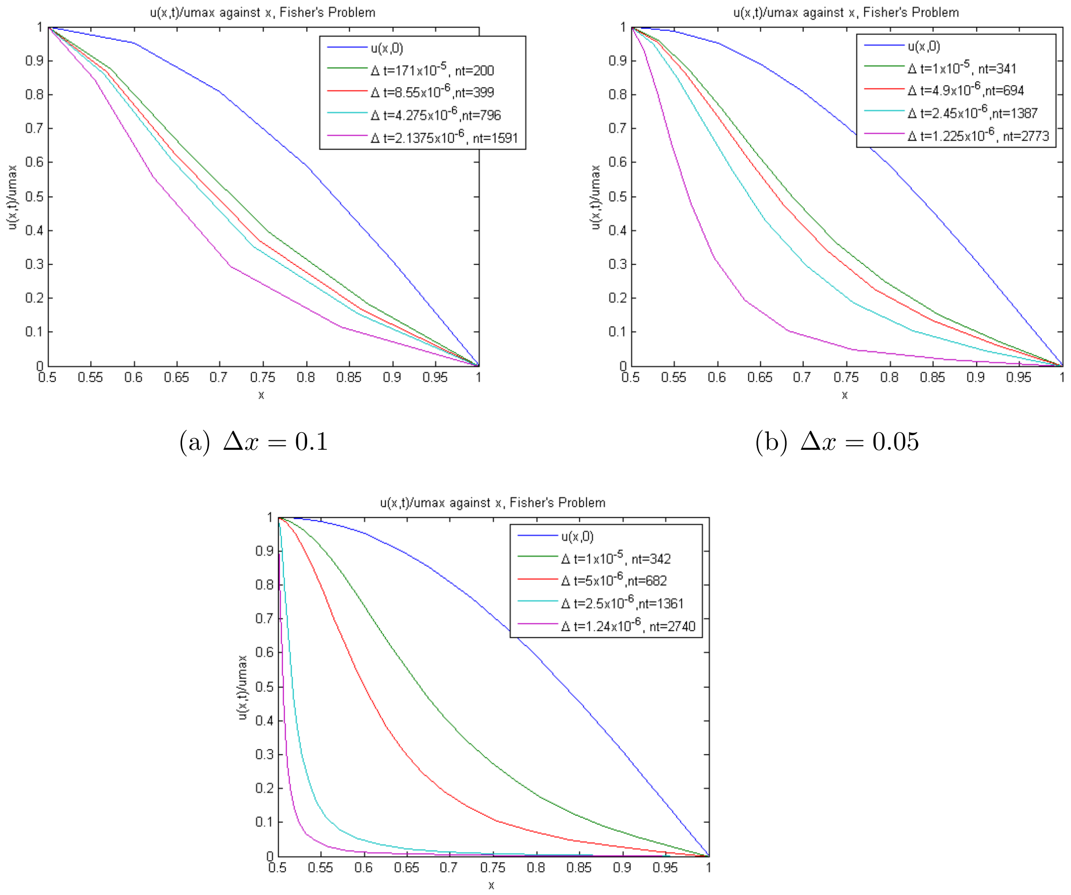
(a) $\Delta x = 0.1$

(b) $\Delta x = 0.05$

(c) $\Delta x = 0.025$

Figure 4.1: Blow-up of $u(x, t)$

For ease of viewing we can normalise the solution range, $u(x,t)$ to $[0,1]$.



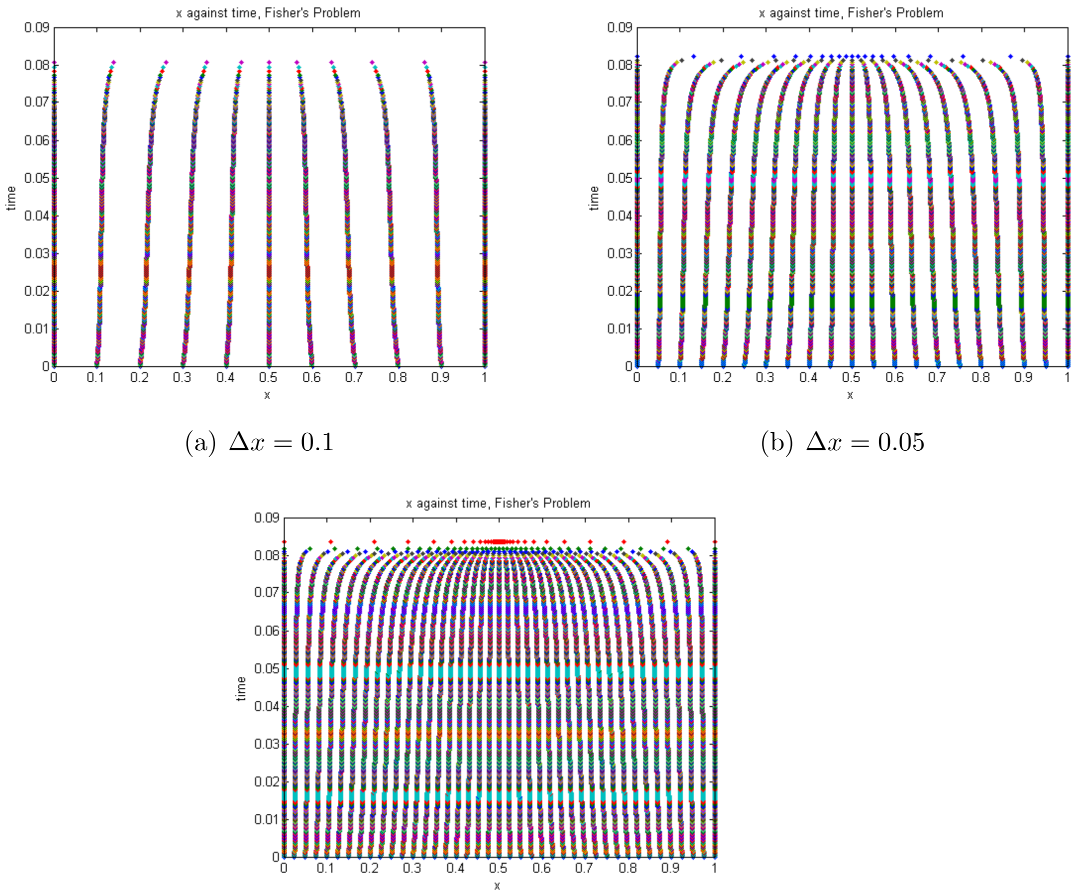
(a) $\Delta x = 0.1$

(b) $\Delta x = 0.05$

(c) $\Delta x = 0.025$

Figure 4.2: Normalisation of the blow-up $u(x,t)$

We can see that by having a small Δt_0 and small Δx, how the solution converges faster. In Figure(7.1 c) the solution is tending towards a Dirac delta function faster than the results in Figure(7.1a and b)



(a) $\Delta x = 0.1$

(b) $\Delta x = 0.05$

(c) $\Delta x = 0.025$

Figure 4.3: Nodes moving in towards the singularity

The images in Figure(4.3) represent the entire domain, $x \in [0, 1]$. We can see how the nodes are moving in towards the singularity which occurs at $x^* = 0.5$. In Figure (4.3,c), we can see how the nodes begin to move extremely fast towards $x^* = 0.5$ on the last iteration, just before T approaches 0.082372.

Since we have been able to approach the blow-up time $T \approx 0.082372$ stated in [2], it would appear that our numerical method is less computationally expensive than the method in [2].

We now turn our attention to investigations carried out on the chemotaxis model.

Chapter 5

A Previous Numerical Investigation of Chemotaxis

We have studied the chemotaxis paper by Budd et al, [1] where they wish to achieve numerical cmputation of the blow-up time. Budd et al, use a method of equidistribution to re-space the nodes. this is achieved by using a moving mesh PDE (MMPDE) and a monitor function.

One of the features of Budd's paper is the scale invariance of the chemotaxis equations, and Budd et al use this idea in relation to the local behaviour of the blow-up.

5.1 Self-Similarity

We first describe the notion of similarity, which will play a part in later theory. For any PDE connecting the variables u,x and t, if we take a scaling of these values,

$$u \rightarrow u' \ = \ \lambda^{\gamma} u$$

$$\begin{aligned} x \longrightarrow x' &= \lambda^{\alpha} x \\ t \longrightarrow t' &= \lambda t \end{aligned}$$

and if for certain α, γ the PDE is invariant, then it is said to be scale-invariant. The variables $\frac{x}{t^{\alpha}}$ and $\frac{u}{t^{\gamma}}$ are independent of λ and are called similarity variables. Also there may be special 'self-similar' solutions of the form

$$\frac{u}{t^{\gamma}} = f\left(\frac{x}{t^{\alpha}}\right)$$

see (5.1).

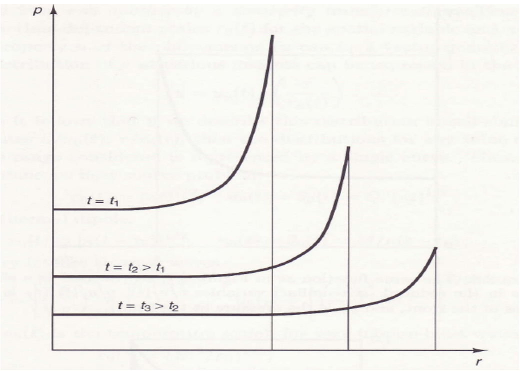


Figure 5.1: Self-Similarity, from [5]

In [1] it is stated that "In the case of true self-similar blow-up (where the solution is invariant under the scaling laws which apply to the differential equation) we expect to see the simpler power law relation
$u(0,t)$ $(T-t)^{-1}$, $L(t)$ $(T-t)^{0.5}$". Hence in our case we will take $\gamma = -1$ and $\alpha = 0.5$.

Budd et al [1], also state that for the one spatial dimension case of a set of the chemotaxis equations with no chemical decay and no chemical diffusion, the blow-up profile can be shown to be self-similar, "with the cell concentration tending to a Dirac-delta function with 'height' inversely proportional to the time to blow-up”[1].

However, for the two spatial dimension radial case solutions have only approximate self-similar behaviour, where the solution does not obey the scaling laws of the PDE. For the three spatial dimension radial case the blow-up occurs in a self-similar way, therefore obeying a strict power law.

5.2 Moving mesh PDE

Budd et al use a remeshing method. The mesh points are re-allocated by solving a moving mesh PDE (MMPDE), which is based on the ideas in [7]. Spatial mesh movement is done based upon equidistribution of a monitor function. The monitor function is chosen so that it is suitable for the numerical solution to resolve the detail in the asymptotic solution. This way the computations are consistent with the asymptotic description of the blow up [1]. However, when choosing the monitor function a compromise needs to be made between obtaining a fine enough mesh to resolve the solution and avoiding stiffness in the solution [1].

For the MMPDE method the physical PDEs are discretised in space on a non-uniform mesh using a ‘cubic Hermite collocation-type method’, as explained in [8]. The MMPDEs are discretised in computational space using a 3-point finite difference method.

The MMPDE used in Budd’s paper is referred to as **MMPDE6**, which is

$$-X_{\xi\xi t} = \frac{1}{\tau}(M(u)X_\xi)_\xi$$

where $M(u)$ is the monitor function.

Once MMPDE6 is solved in computational space, the nodes in physical space should move so that they accumulate in the area that the singularity occurs and the solution can be tracked in physical space, as explained in [7].

5.3 Choosing a suitable Monitor Function

According to [1], when choosing a suitable monitor function, several elements need to be taken into consideration. When updating the mesh points close to blow-up time, T, it is essential the nodes do not move too fast, otherwise the resulting system of equations is very stiff. The nodes must not move too slowly, otherwise the blow-up will not be tracked. They should evolve at the same rate as the underlying solution. In order to evolve the mesh points at the correct time-scale a compromise between the above points need to be made. It is stated in the paper that the monitor function needs to satisfy

$$M(u) \propto (T-t)^{-1}$$

asymptotically, which for the chemotaxis problem can be achieved by using a monitor function of the form

$$M(u) = u^{\beta}$$

For the 2D Chemotaxis case Budd et al in [1] use

$$M(u) = u^{\frac{1}{2}}$$

and for the 3D case

$$M(u) = u$$

5.4 Blow-up Time

Budd et al [1] investigate the Keller-Segel model with initial conditions stated in Chapter 1. It is shown that the solution becomes singular when $T \approx 5.15 \times 10^{-5}$. Figure (5.2) is taken from [1] and based on [6]. It shows the general form of the solution u in the chemotaxis problem as it blows up.

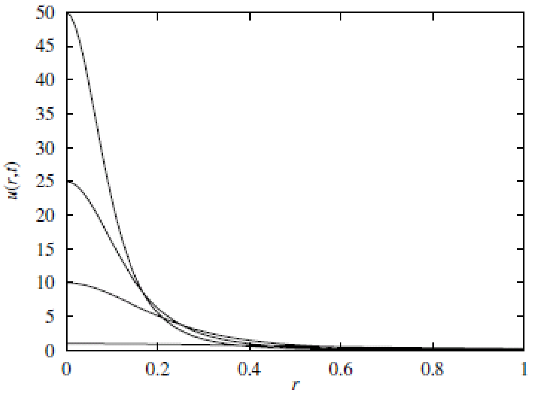


Figure 5.2: The behaviour of u in the core region for varying γ, from [1]

Although Budd et al use the method of equidistribution in [1], for the chemotaxis problem, we will use a velocity based method as mentioned in Chapter 3. The mass of each element under the solution curve will be conserved.

Chapter 6

Method of conservation for the Chemotaxis Model

For the chemotaxis equations, we will be looking only at the 2D and 3D radial cases.

6.1 Chemotaxis 2D

For the 2D case the radially symmetric Keller-Segel system of equations (1.1) in cylindrical polar co-ordinates becomes

$$\begin{aligned} u_t &= \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right) - \chi\frac{1}{r}\frac{\partial}{\partial r}\left(ru\frac{\partial v}{\partial r}\right) \\ v_t &= \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial v}{\partial r}\right) + u - v \end{aligned}$$

in (0,R). As in [1] the initial conditions are taken to be

$$u(r,0) = 1000e^{-500r^2}$$

$$v(r,0) \quad = \quad 10e^{-500r^2}$$

We note from [1] that

$$\int_0^R urdr = \text{constant} \tag{6.1}$$

From (6.1), and the initial conditions, we can generate the velocities of the moving nodes.

6.1.1 Generating Velocities

We shall assume that

$$\int_{r_{i-1}(t)}^{r_i(t)} urdr = \text{constant}$$

which is obviously consistent with (6.1). Then by Leibniz's rule

$$\begin{aligned}
\frac{d}{dt}\int_{r_{i-1}(t)}^{r_i(t)} urdr &= 0 \\
&= \int_{r_{j-1}(t)}^{r_i(t)} ru_t dr + [rus]_{r_{i-1}(t)}^{r_i(t)} \\
&= \int_{r_{j-1}(t)}^{r_i(t)} \left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right) - \chi\frac{1}{r}\frac{\partial}{\partial r}\left(ru\frac{\partial v}{\partial r}\right)\right] rdr + [rus]_{r_{i-1}(t)}^{r_i(t)} \\
&= \left[r\frac{\partial u}{\partial r} - \chi ru\frac{\partial v}{\partial r} + rus\right]_{r_{i-1}(t)}^{r_i(t)}
\end{aligned}$$

Summing for all i from 0 to j we obtain

$$\left[r\frac{\partial u}{\partial r} - \chi r u \frac{\partial v}{\partial r} + rus\right]_0^j = 0$$

so

$$\begin{aligned} 0 &= r_j(u_r)_j - \chi r_j u_j (v_r)_j + r_j u_j s_j \\ &= (u_r)_j - \chi u_j (v_r)_j + u_j s_j \\ \Rightarrow u_j s_j &= -(u_r)_j + \chi u_j (v_r)_j \\ \Rightarrow s_j &= -\frac{(u_r)_j}{u_j} + \chi (v_r)_j \end{aligned}$$

6.1.2 Core Region

Since the singularity forms close to $r = 0$, we are going to consider the domain [0,0.08],(which we will refer to as the core region), rather than [0,1]. At $t = 0$ we have

$$u = 1000e^{-500r^2}, \qquad v = 10e^{-500r^2}$$

so the velocity of the nodes at this time will be

$$\begin{aligned} s(0) &= -\frac{\partial}{\partial r}\left[\ln 1000 - 500r^2 - \chi 10 e^{-500r^2}\right] \\ &= 1000r\left[1 - 10\chi e^{-500r^2}\right] \end{aligned}$$

The singularity occurs close to $r = 0$, when moving the nodes we want them to approach this area, so that the singularity can be refined. Therefore the nodes will be moving in from the right boundary into $r = 0$. This means that we require the velocity of the nodes to be negative. So we want

$$s(0) < 0$$

so we have

$$\begin{aligned} s(0) &= 1000r\left[1 - 10\chi e^{-500r^2}\right] < 0 \\ \Rightarrow 1 &< 10\chi e^{-500r^2} \end{aligned}$$

since $\chi = 8$ we have

$$e^{-500r^2} > \frac{1}{80}$$

If we take $r \in [0, 0.08]$, this inequality is satisfied.

6.1.3 Recovering New r,u and v Values

Now that we know the velocity of the nodes we can calculate their new position. The change in r_j with respect to time is

$$\frac{dr_j}{dt} = s_j$$

By using the Euler forward timestepping method, we have

$$r_j^{n+1} = r_j^n + \Delta t s_j$$

From the r_j's new values of u can be calculated. Since the area of the individual regions are conserved in time, we have (combining two of the integrals)

$$\int_{r_{j-1}}^{r_{j+1}} urdr$$

is constant in time, equal to its initial value which means

$$\int_{r_{j-1}(t)}^{r_{j+1}(t)} u(t)r(t)dr = \int_{r_{j-1}(0)}^{r_{j+1}(0)} u(0)r(0)$$

so by a mid-point rule

$$\begin{aligned} r_j(t)\left[r_{j+1}(t) - r_{j-1}(t)\right]u_j(t) &= r_j(0)\left[r_{j+1}(0) - r_{j-1}(0)\right]u_j(0) \\ \Rightarrow u_j(t) &= \frac{r_j(0)\left[r_{j+1}(0) - r_{j-1}(0)\right]u_j(0)}{r_j(t)\left[r_{j+1}(t) - r_{j-1}(t)\right]} \end{aligned}$$

We can also calculate the new values of v. We have that

$$v_t = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial v}{\partial r}\right) + u - v$$

in cylindrical polar co-ordinates, but on the moving mesh the new values of v will have an extra term added due to the movement of the mesh, by the chain rule. The modified equation is

$$\begin{aligned} \frac{dv}{dt} &= \frac{1}{r}\left(rv_r\right)_r + u - v + sv_r && (6.2) \\ & && (6.3) \\ &= v_{rr} + \frac{1}{r}v_r + u - v + sv_r \end{aligned}$$

where $\frac{dv}{dt}$ is the rate of change of v in the moving frame, which can be discretised in space and time to give

$$\frac{v_j^{n+1} - v_j^n}{\Delta t} = \frac{(v_{j-1}^n - 2v_j^n + v_{j+1}^n)}{\Delta r^2} + \left(\frac{1}{r} + s_j^n\right)\left(\frac{v_{j+1}^n - v_{j-1}^n}{2\Delta r}\right) + u_j^n - v_j^n$$

$$\Rightarrow v_j^{n+1} = v_j^n + \Delta t\left[\frac{(v_{j-1}^n - 2v_j^n + v_{j+1}^n)}{\Delta r^2} + \left(\frac{1}{r} + s_j^n\right)\left(\frac{v_{j+1}^n - v_{j-1}^n}{2\Delta r}\right) + u_j^n - v_j^n\right] \tag{6.4}$$

We can use (6.4) for interior nodes, but on the left and right boundaries (6.4) needs to be modified.

Left boundary, $j = 1$

Since we have Neumann boundary conditions, $v_r = 0$, we introduce a ghost point r_{-1}, and apply (6.2) at $r = 0$ with

$$\frac{v_1 - v_{-1}}{2\Delta r} = 0$$

$$\Rightarrow v_{-1} = v_1$$

However, difficulties arise in applying (6.4) at $r = 0$. Instead we use the approximation

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial v}{\partial r}\right) = 2\frac{\partial^2 v}{\partial r^2}$$

see [3] for a further explanation on this.

Now we have the following equation at the left hand boundary,

$$v_t = 2v_{rr} + u - v + sv_r$$

and, after discretising in time and space we obtain

$$\frac{v_0^{n+1} - v_0^n}{\Delta t} = \left(2\frac{v_{-1}^n - 2v_0^n + v_1^n}{\Delta r^2}\right) + u_0^n - v_0^n + s_0^n\left(\frac{v_0^n - v_{-1}^n}{\Delta r}\right)$$

since we have $v_{-1} = v_1$ we get

$$v_0^{n+1} = v_0^n + \Delta t\left[\left(4\frac{v_1^n - v_0^n}{\Delta r^2}\right) + u_0^n - v_0^n + s_0^n\left(\frac{v_1^n - v_0^n}{\Delta r}\right)\right]$$

Right boundary, $j = nr$

Since our domain is $r \in [0, 0.08]$ we cannot use the Neumann boundary condition on its own, since this is only valid at $r = 1$.

From the initial conditions we have

$$\begin{aligned} v &= 10e^{-500r^2} \\ \Rightarrow v_t &= -10000re^{-500r^2} \\ \Rightarrow 0 &= v_r + 1000rv \end{aligned}$$

which is a Robin boundary condition. Since $r = 0.08$ on this boundary we have

$$v_r = -80v$$

which can now be discretised in space to give

$$\begin{aligned} \frac{v_{nr+1} - v_{nr-1}}{2\Delta r} &= -80v_{nr} \\ \Rightarrow v_{nr+1} &= v_{nr-1} - 160\Delta r v_{nr} \qquad (6.5) \end{aligned}$$

Since $r \neq 0$ we do not have the difficulties in approximating the derivatives with standard finite differences, so we have

$$v_{nr}^{n+1} = v_{nr}^{n} + \Delta t \left[\frac{(2v_{nr-1}^{n} - 2v_{nr}^{n} - 160\Delta r v_{nr})}{\Delta r^2} + u_n r^n - v_n r^n + \left(s_{nr}^{n} + \frac{1}{r_{nr}^{n}} \right) \left[\frac{v_{nr}^{n} - v_{nr-1}^{n}}{\Delta r} \right] \right] \qquad (6.6)$$

6.2 Chemotaxis 3D

We now turn our attention to the 3D case. The radially symmetric Keller-Segel system of equations in spherical polar co-ordinates becomes

$$\begin{aligned} u_t &= \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial u}{\partial r}\right) - \chi\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 u\frac{\partial v}{\partial r}\right) \\ v_t &= \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial v}{\partial r}\right) + u - v \end{aligned} \tag{6.7}$$

again with initial conditions

$$\begin{aligned} u(r,0) &= 1000e^{-500r^2} \\ v(r,0) &= 10e^{-500r^2} \end{aligned}$$

By construction,

$$\int_0^R ur^2 dr = \text{constant}$$

which gives us

$$\frac{d}{dt}\int_0^R ur^2 dr = 0 \tag{6.8}$$

This allows us to calculate the velocities of the moving nodes, and in turn we will be able to recover new r,v and u values.

6.2.1 Generating Velocities

We assume that for all i

$$\frac{d}{dt}\int_{r_{i-1}}^{r_i} ur^2 dr = 0$$

$$
\begin{aligned}
&= \int_{r_{i-1}}^{r_i} r^2 u_t dr + \left[r^2 us\right]_{r_{i-1}}^{r_i} \qquad (6.9)\\
&= \int_{r_{i-1}}^{r_i} \left[\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial u}{\partial r}\right) - \frac{1}{r^2}\chi\frac{\partial}{\partial r}\left(r^2 u\frac{\partial v}{\partial r}\right)\right] r^2 dr + \left[r^2 us\right]_{r_{i-1}}^{r_i}\\
&= \left[r^2\frac{\partial u}{\partial r} - \chi r^2 u\frac{\partial v}{\partial r} + r^2 us\right]_{r_{i-1}}^{r_i}
\end{aligned}
$$

Summing for all i from 0 to j we obtain

$$\left[r^2\frac{\partial u}{\partial r} - \chi r^2 u\frac{\partial v}{\partial r} + r^2 us\right]_0^j = 0$$

so

$$
\begin{aligned}
0 &= (r_j)^2(u_r)_j - \chi(r_j)^2 u_j(v_r)_j + (r_j)^2 u_j s_j\\
&= (u_r)_j - \chi u_j(v_r)_j + u_j s_j\\
\Rightarrow u_j s_j &= -(u_r)_j + \chi u_j(v_r)_j\\
\Rightarrow s_j &= -\frac{(u_r)_j}{u_j} + \chi(v_r)_j
\end{aligned}
$$

From this and Section(6.1.2) we again compute on the domain $r \in [0, 0.08]$.

6.2.2 Recovering New r,u and v Values

Now that we know the velocity of the nodes we can calculate their new position. The change in r_j with respect to time is

$$\frac{dr_j}{dt} = s_j$$

by using an Euler timestepping method, we have

$$r_j^{n+1} = r_j^n + \Delta t s_j$$

From this the new values of u can be calculated. Since the area of the individual regions are conserved in time, equal to its initial value, we have (combining two of the integrals)

$$\int_{r_{j-1}}^{r_{j+1}} ur^2 dr$$

is constant in time, which means

$$\begin{aligned} \int_{r_{j-1}(t)}^{r_{j+1}(t)} u(t)r^2(t)dr &= \int_{r_{j-1}(0)}^{r_{j+1}(0)} u(0)r^2(0)dr \\ \Rightarrow (r_j(t))^2 \left[r_{j+1}(t) - r_{j-1}(t)\right] u_j(t) &= (r_j(0))^2 \left[r_{j+1}(0) - r_{j-1}(0)\right] u_j(0) \\ \Rightarrow u_j(t) &= \frac{(r_j(0))^2 \left[r_{j+1}(0) - r_{j-1}(0)\right] u_j(0)}{(r_j(t))^2 \left[r_{j+1}(t) - r_{j-1}(t)\right]} \end{aligned}$$

We can also calculate the new values of v. We have (6.7) that

$$v_t = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 \frac{\partial v}{\partial r}\right) + u - v$$

The new values of v will have an extra term added due to the movement of the mesh by the chain rule. This is given by

$$\begin{aligned}\frac{dv}{dt} &= \frac{1}{r^2}\left(r^2 v_r\right)_r + u - v + sv_r \\ &= v_{rr} + \frac{2}{r}v_r + u - v + sv_r\end{aligned}$$

which can be discretised in space and time to give

$$\frac{v_j^{n+1} - v_j^n}{\Delta t} = \frac{(v_{j-1}^n - 2v_j^n + v_{j+1}^n)}{\Delta r^2} + \left(\frac{2}{r} + s_j^n\right)\left(\frac{v_{j+1}^n - v_{j-1}^n}{2\Delta r}\right) + u_j^n - v_j^n$$

$$\Rightarrow v_j^{n+1} = v_j^n + \Delta t\left[\frac{(v_{j-1}^n - 2v_j^n + v_{j+1}^n)}{\Delta r^2} + \left(\frac{2}{r} + s_j^n\right)\left(\frac{v_{j+1}^n - v_{j-1}^n}{2\Delta r}\right) + u_j^n - v_j^n\right] \tag{6.10}$$

We can use (6.10) for interior nodes, but on the left and right boundaries (6.10) needs to be modified.

Left boundary, $j = 1$

Since we have Neumann boundary conditions, $v_r = 0$, we introduce a ghost point r_{-1} and apply (6.7) at $r = 0$ with

$$\begin{aligned}\frac{v_1 - v_{-1}}{2\Delta r} &= 0 \\ \Rightarrow v_{-1} &= v_1\end{aligned}$$

However, difficulties arise in applying (6.10) at $r = 0$. Instead we use the approximation

$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial v}{\partial r}\right) = 3\frac{\partial^2 v}{\partial r^2}$$

see [3].

Now we have the following equation at the left hand boundary

$$v_t = 3v_{rr} + u - v + sv_r$$

and, after discretisation in time and space we obtain

$$\frac{v_0^{n+1} - v_0^n}{\Delta t} = \left(3\frac{v_{-1}^n - 2v_0^n + v_1^n}{\Delta r^2}\right) + u_0^n - v_0^n + s_0^n\left(\frac{v_0^n - v_{-1}^n}{\Delta r}\right)$$

Since $v_{-1} = v_1$ we get

$$v_0^{n+1} = v_0^n + \Delta t\left[\left(6\frac{v_1^n - v_0^n}{\Delta r^2}\right) + u_0^n - v_0^n + s_0^n\left(\frac{v_1^n - v_0^n}{\Delta r}\right)\right]$$

Right boundary, $j = nr$

Since our domain is $r \in [0, 0.08]$ we cannot use the Neumann boundary condition on its own, since this is only valid at r=1. From Section(6.1.3) we approximate v_{nr+1} by equation (6.5).

Since $r \neq 0$ we do not have the difficulties in approximating the derivatives in (6.7) with standard finite differences [3], so we have

$$\begin{aligned} v_{nr}^{n+1} &= v_{nr}^n + \Delta t\left[\frac{(2v_{nr-1}^n - 2v_{nr}^n - 160\Delta r v_{nr})}{\Delta r^2} + \right. \\ &+ \left. u_n r^n - v_n r^n + \left(s_{nr}^n + \frac{2}{r_{nr}^n}\right)\left[\frac{v_{nr}^n - v_{nr-1}^n}{\Delta r}\right]\right. \end{aligned} \tag{6.12}$$

Chapter 7

Chemotaxis Results

7.1 Chemotaxis 2D

We considered the 2D chemotaxis problem and looked at how the solution was affected by

- an increase in spatial nodal values, (nr)
- decreasing the timestep, (Δt)
- increasing the number of timesteps,(nt), as the size of Δt decreases

7.1.1 Changing Δt, nr and nt

We looked at the results for $u(r, t)$, $v(r, t)$ and the change in r to see how they were affected by taking different combinations of Δt, nr and nt.

Table 7.1: Changing Δt, nr and nt for the 2D chemotaxis problem

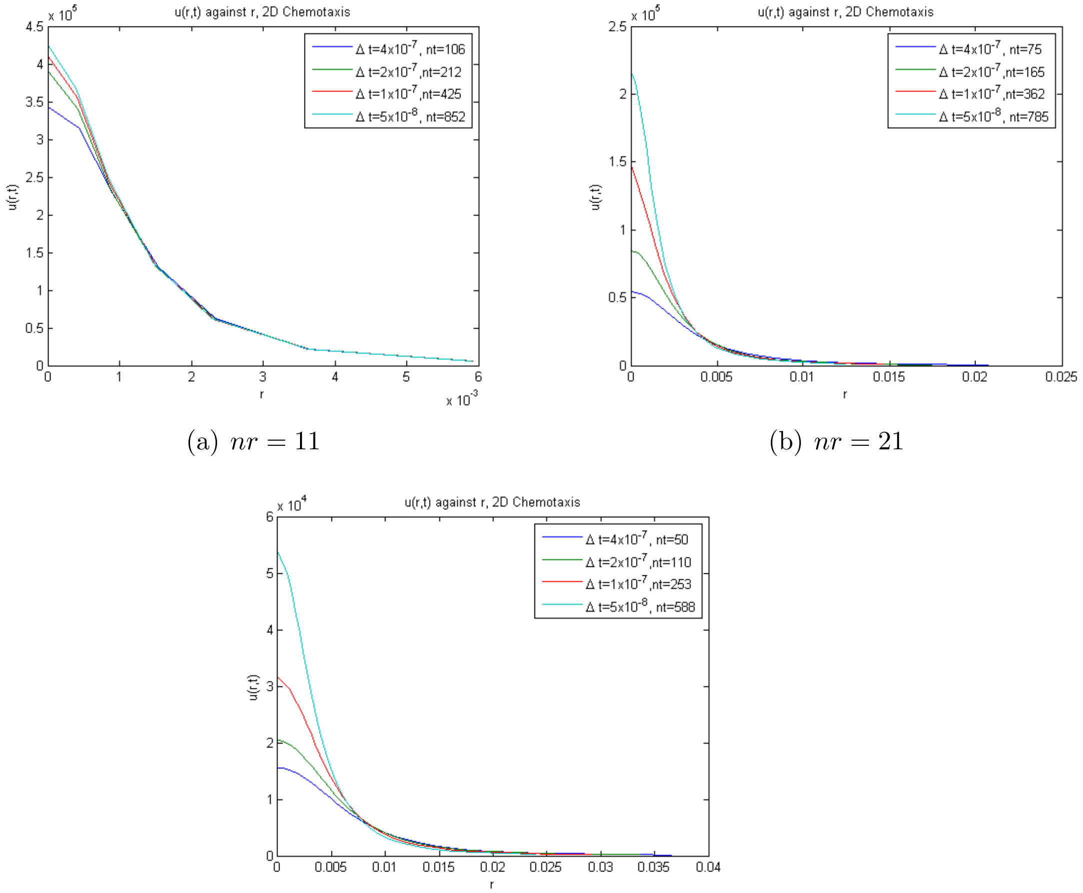
nr	Δt	nt	T
11	4×10^{-7}	106	4.24×10^{-5}
11	2×10^{-7}	212	4.24×10^{-5}
11	1×10^{-7}	425	4.25×10^{-5}
11	5×10^{-8}	852	4.26×10^{-5}
21	4×10^{-7}	75	3×10^{-5}
21	2×10^{-7}	165	3.3×10^{-5}
21	1×10^{-7}	362	3.62×10^{-5}
21	5×10^{-8}	785	3.925×10^{-5}
41	4×10^{-7}	52	2.08×10^{-5}
41	2×10^{-7}	112	2.24×10^{-5}
41	1×10^{-7}	253	2.53×10^{-5}
41	5×10^{-8}	588	2.94×10^{-5}

From Table (7.1) we can see that by fixing Δr and decreasing Δt we can take more timesteps. With this the blow up time increases.

By increasing nr and keeping Δt we need fewer timesteps for blow-up. However, the blow-up time is decreasing.

7.1.2 Solution of $u(r,t)$

To begin with we are interested in the evolution of the cell density,$u(r,t)$, throughout the time period. Although the numerical calculations were carried out on $r \in [0, 0.08]$, we have only plotted for a smaller region as to see a refined version of the solution. This is shown in Figure(7.1)



(a) $nr = 11$

(b) $nr = 21$

(c) $nr = 41$

Figure 7.1: Blow-up of $u(r,t)$, for a 2D Chemotaxis System

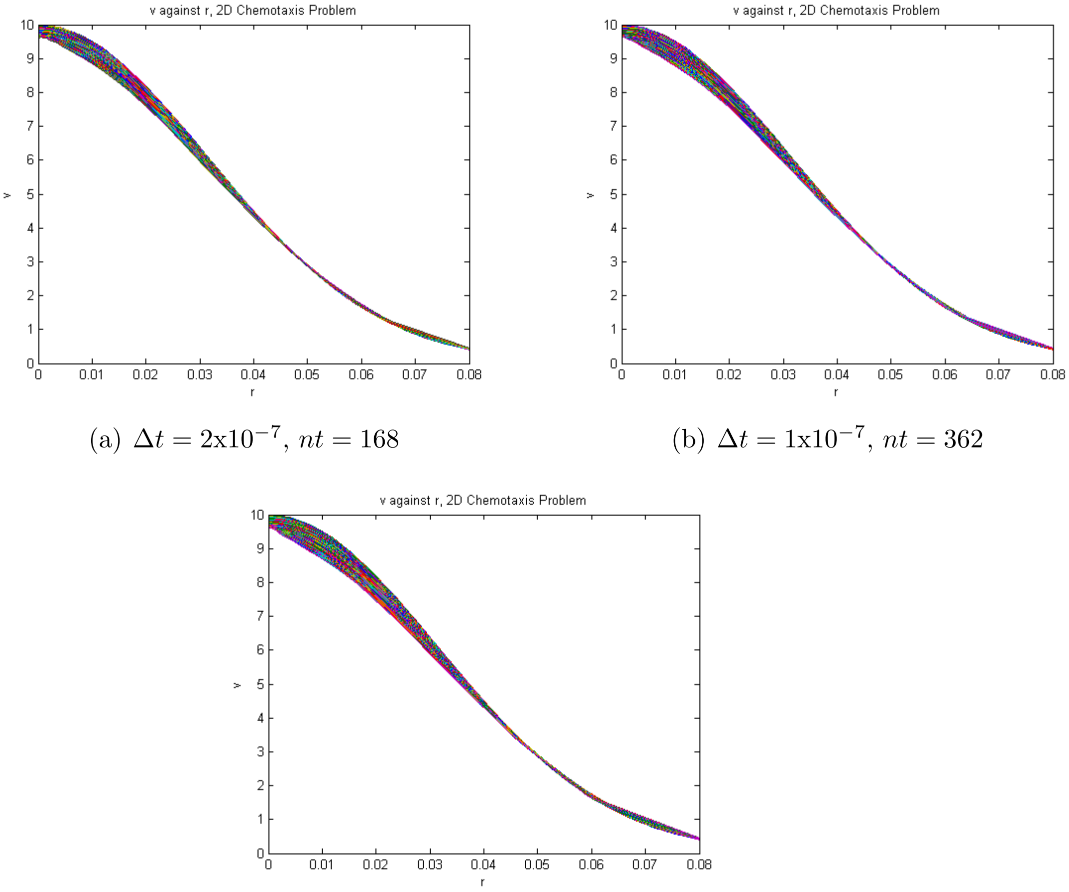
From Figure(7.1) we can see in all figures that as we approach $r = 0$ the solution begins to blow-up. As the size of the timestep Δt is decreased and the number of timesteps nt is increased, the solution will blow-up more. This is true for all cases in Figure(7.1).

As we increase the number of spatial nodes nr, the overall blow-up becomes less. We can see in Figure(7.1.a) where $nr = 11$ that the solution blows-up to 4×10^6. In Figure(7.1.b) where $nr = 21$ the solution blows-up to 2.5×10^4.

An advantage of using more nodes is the refinement away from the blow-up. Although the nodes are moving in towards the singularity, when the number of nodes is increased we can obtain more information away from the blow-up, than when the number of spatial nodes is decreased.

7.1.3 Solution of $v(r,t)$

As the cell density $u(r,t)$ blows up over time, we are interested to see what role the the concentration of chemical substrate $v(r,t)$ has in the blow-up of the solution. We look at the solution of $v(r,t)$ when there are 21 nodes.



(a) $\Delta t = 2\text{x}10^{-7}$, $nt = 168$

(b) $\Delta t = 1\text{x}10^{-7}$, $nt = 362$

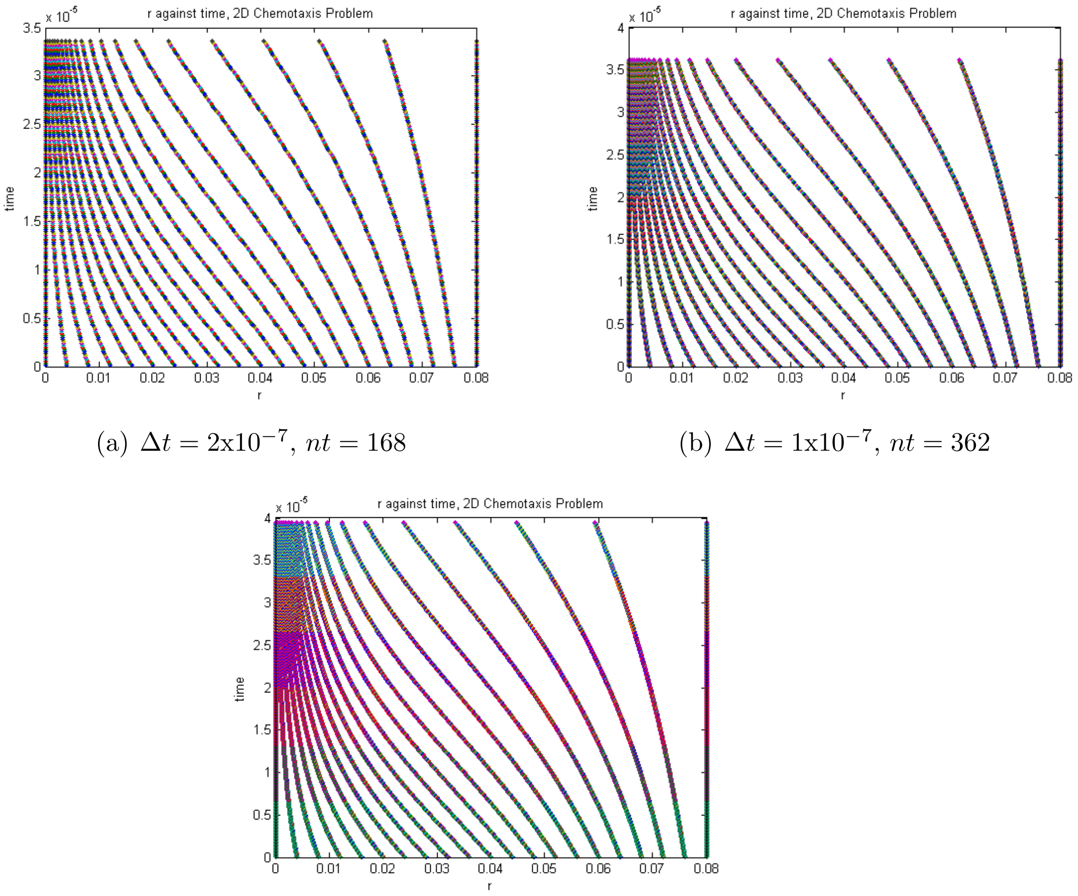
(c) $\Delta t = 5\text{x}10^{-8}$, $nt = 789$

Figure 7.2: Solution of $v(r,t)$, when $nr = 21$, for a 2D Chemotaxis System

There is barely any change in the solution of $v(r,t)$ through the evolution. This suggests that the concentration of the chemical substrate is remaining fairly constant in the environment. The importance of the concentration of the chemical substrate remaining fairly constant, and the effect it has on the blow up of $u(r,t)$ will be left for further work.

7.1.4 Movement of the Nodes

The main aim of this project is to move the spatial nodes so that any singularities in the solution can be followed and refined. We look at the movement of the nodes when $nr = 21$



(a) $\Delta t = 2\text{x}10^{-7}$, $nt = 168$

(b) $\Delta t = 1\text{x}10^{-7}$, $nt = 362$

(c) $\Delta t = 5\text{x}10^{-8}$, $nt = 789$

Figure 7.3: Movement of the nodes,r,throughout time when $nr = 21$, for a 2D Chemotaxis System

From Figure(7.3) we can see how the nodes move in towards the singularity at $r = 0$. In Figure(7.3.a) the nodes are not as clustered around $r = 0$ as they are in Figure(7.3.c). However, we can see in Figure(7.3.c) that throughout the evolution as the nodes get closer and closer to the singularity, there are fewer nodes away from the blow-up point. This means that if there is any interesting behaviour in this area then it will not necessarily be detected.

7.1.5 A Numerical Self-Similarity Property

For self-similar behaviour Budd et al [1] expects the power law relation to be $L(t)$ $(T-t)^{0.5}$. Although Budd states that the 2D case is "'not strictly self-similar"'[1], we found that in our results $\frac{r}{t^\beta}$ was eventually almost self-similar.

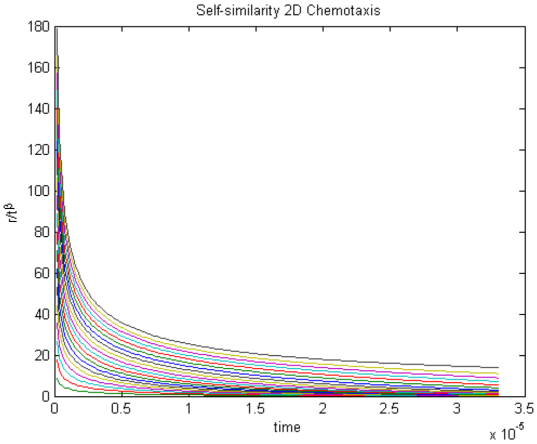


Figure 7.4: Self-similarity of $\frac{r}{t^\beta}$ for 2D Chemotaxis, where $\Delta t = 2 \times 10^{-7}$, $nt = 165$ and $nr = 21$

7.2 Chemotaxis 3D

7.2.1 Changing Δt, nr and nt

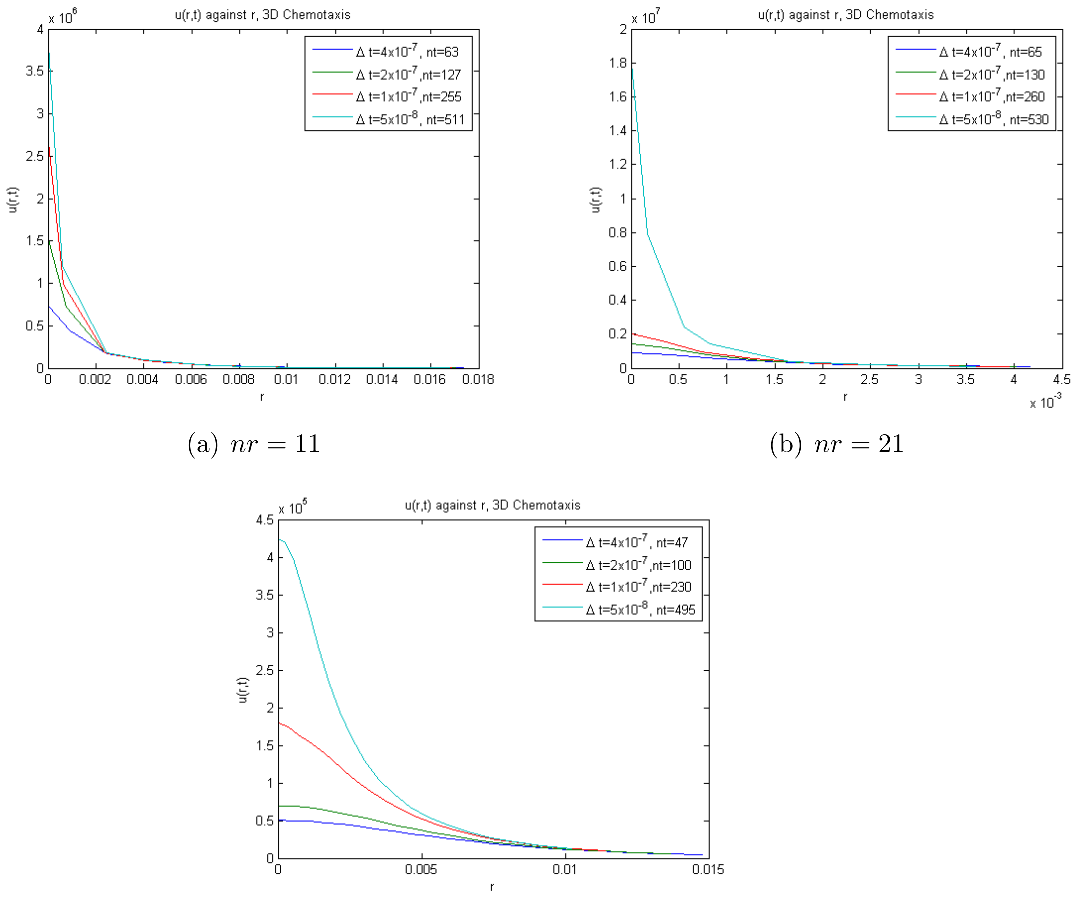
From Table(7.2) we can see that by fixing Δr and decreasing Δt we can take more timesteps. With this the blow up time increases. By increasing nr and keeping Δt we need fewer timesteps for blow-up. However, the blow-up time is decreasing.

Table 7.2: Changing Δt, nr and nt for the 3D chemotaxis problem

nr	Δt	nt	T
11	4×10^{-7}	63	2.52×10^{-5}
11	2×10^{-7}	127	2.54×10^{-5}
11	1×10^{-7}	255	2.55×10^{-5}
11	5×10^{-8}	511	2.555×10^{-5}
21	4×10^{-7}	65	2.6×10^{-5}
21	2×10^{-7}	130	2.6×10^{-5}
21	1×10^{-7}	260	2.6×10^{-5}
21	5×10^{-8}	530	2.65×10^{-5}
41	4×10^{-7}	47	1.88×10^{-5}
41	2×10^{-7}	100	2×10^{-5}
41	1×10^{-7}	230	2.3×10^{-5}
41	5×10^{-8}	495	2.47×10^{-5}

7.2.2 Solution of $u(r,t)$

We will again look at the evolution of the cell density, $u(r,t)$, throughout the time period as we approach a blow up solution. In (7.5) we can see the solution blow-up as we approach $r = 0$, as we did in the 2D case. Once again we can observe that the blow-up of $u(r,t)$ increases by increasing nt and decreasing Δt.



(a) $nr = 11$

(b) $nr = 21$

(c) $nr = 41$

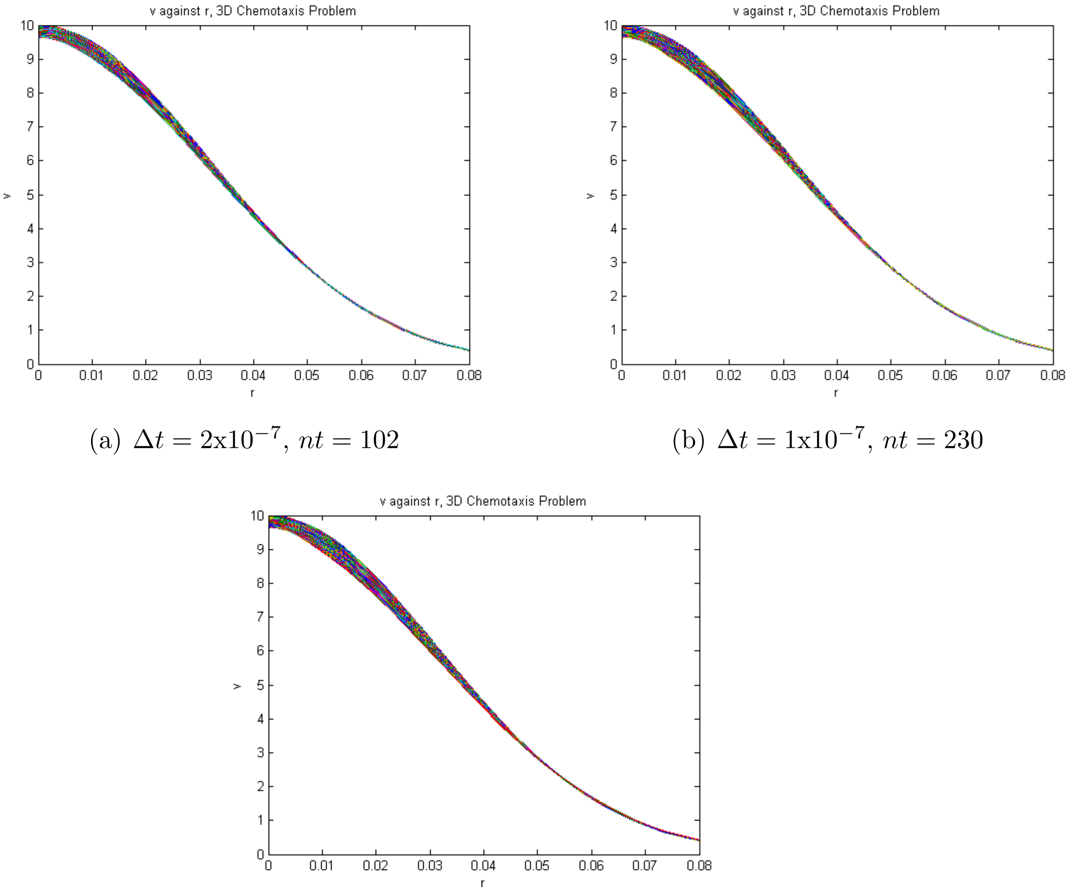
Figure 7.5: Blow-up of $u(r,t)$, for a 3D Chemotaxis System

In the 3D case, it takes fewer timesteps (for the same Δt in the 2D case), for the solution to blow up, although the solution does not blow-up as much.

By increasing the number of spatial nodes (nr), we see the maximum that $u(r,t)$ reaches is higher in (7.5.b), where nr has increased. However, $u(r,t)$ has a smaller maximum in (7.5.c), where the nx is greater than (7.5a and b)

7.2.3 Solution of $v(r, t)$

We again want to see the effect that the concentration of the chemical substrate $v(r, t)$, has on the cell density $u(r, t)$, as the solution blows up. We look at $v(r, t)$, when we have 41 spacial nodes. We see in Figure(7.6), that $v(r, t)$ hardly changes throughout the evolution. An increase in spatial nodes has no effect on this solution. This indicates that it is an important feature to allow $u(r, t)$ to blow-up.



(a) $\Delta t = 2\text{x}10^{-7}$, $nt = 102$

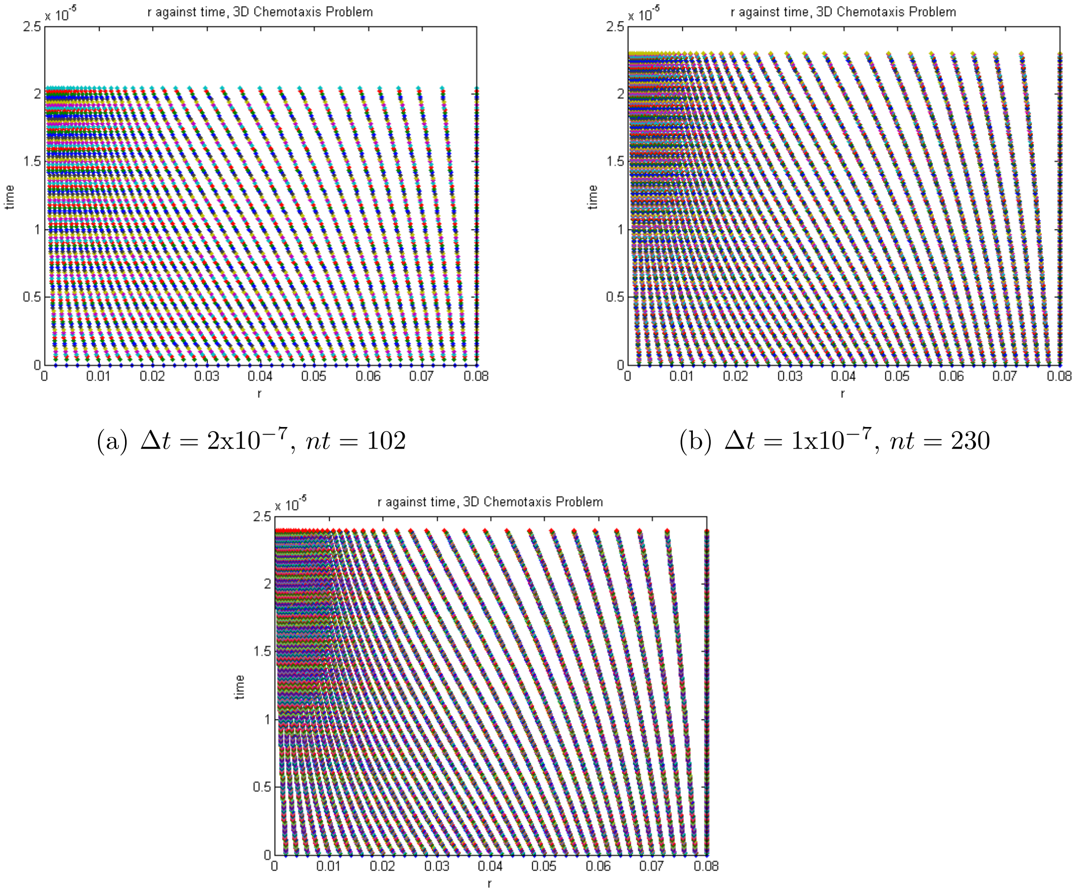
(b) $\Delta t = 1\text{x}10^{-7}$, $nt = 230$

(c) $\Delta t = 5\text{x}10^{-8}$, $nt = 479$

Figure 7.6: Solution of $v(r, t)$, when $nr = 41$, for a 3D Chemotaxis System

7.2.4 Movement of the Nodes

We will look at the nodes moving when $nr = 41$ In Figure(7.7) we can see how 41 spatial nodes move throughout the evolution. After more timesteps we can see how the nodes become concentrated around $r = 0$, as seen in Figure(7.7.c), where we have 479 timesteps. A positive feature is that by using more nodes in Figure(7.7) we can refine the solution away from blow-



(a) $\Delta t = 2\text{x}10^{-7}$, $nt = 102$

(b) $\Delta t = 1\text{x}10^{-7}$, $nt = 230$

(c) $\Delta t = 5\text{x}10^{-8}$, $nt = 479$

Figure 7.7: Movements of the Nodes, r, when $nr = 41$, for a 3D Chemotaxis System

up, which can not be done as well in Figure(7.3)where there are fewer nodes.

7.2.5 A Numerical Self-Similarity Property

For self-similar behaviour Budd et al [1] expects the power law relation to be $L(t)$ $(T-t)^{0.5}$. By using setting $\alpha = 0.5$, we do eventually see an almost self-similar behaviour for the 3D case

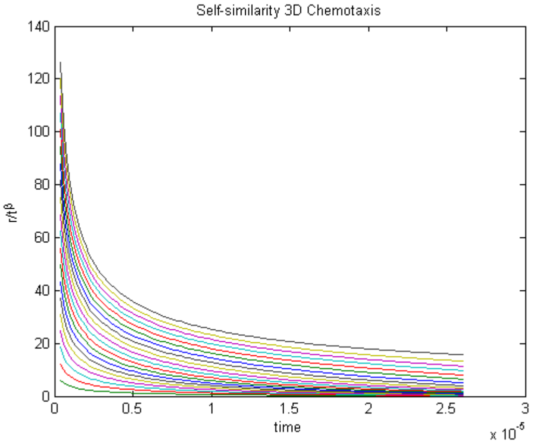


Figure 7.8: Self-similarity of $\frac{r}{t^{\beta}}$ for 3D Chemotaxis, where $\Delta t = 2 \times 10^{-7}$, $nt = 130$ and $nx = 21$.

7.3 Comparison Between 2D and 3D Chemotaxis

The main point to notice from Table(7.1)and(7.2) is that by comparing the solutions where Δr and Δt are the same, we see that the 3D case requires fewer timesteps to blow-up. This means that the blow-up time is less in the 3D case.

We would have expected that in both cases, that as nx is increased, the solution $u(r,t)$ would reach a higher value than the previous nx. However, this has not been the case. An explaination for this could be that as we increase nx the rounding error propogates. Or another possible reason is that numerical diffusion is causing the solution to be smeared out. This could explain why the solution in Figure(7.5) begins to increase as nx increases, but then decreases in Figure(7.5.c).

Budd et al [1] expected to see self-similar behaviour for the 3D chemotaxis case, which we are in agreement with. However, we also observe self-similar behaviour in the 2D case, which is not in line with [1].

Chapter 8

Conclusion

8.1 Discussion of the Project

To begin with we used the Fisher's equation to investigate the existence of blow-up. We found that by using a fixed mesh to numerically compute the solution to the Fisher's equation, that blow-up did exist. However, it became apparent that as the solution reached a higher blow-up point, the fixed mesh would not be able to resolve the solution further. From this we discussed the different types of adaptive mesh procedures and decided a moving mesh method would be useful to resolve our solution as it converged. The moving mesh method used would conserve the relative area underneath the solution curve. In turn we could generate nodal velocities. The nodes could then be moved in towards the blow-up point. This method was applied to the Fisher's equation. With this method we were able to approach the blow-up time that Budd et al had stated in [2], which was $T \approx 0.082372$. We believe that our method is computationally less expensive than the method used in [2].

Our main interest in the problem was to numerically compute the solution to a system of PDEs called the Keller-Segel model. This system is used to model chemotaxis. In [1] this system had been investigated using a moving mesh method. The method in this paper used a MMPDE to determine the position of new nodes, using a monitor function to adequately track the solution. To make comparisons with [1] we used the same model and initial data, but we applied the method of conservation to move the mesh, as was previously described for the Fisher's equation.

We found that as we decreased Δt and increased the number of timesteps the blow-up time increased. However, by increasing nr in many circumstances the value of $u(r,t)$ decreased. There could be many explainations for this, which were discussed in the results chapter. Another explaination for this could have been the Robin bounary conditions we had on the right hand boundary in Chapter(6). Since the solution becomes flat at this boundary we could have used extrapolation to obtain the solution on the boundary. However, it was found that this had no effect on the solution.

8.2 Further Work

To complete this project there are some details which could be investigated in further work. Since we are dealing with a Biological problem it would be interesting to apply realistic numbers to the problem. This way we could make comparisons with experimental data to determine how effective our numerical approach is.

The chemotactic coefficient χ seems to hold a lot of relevance. It would be an idea to investigate how this coefficient effects the overall result in order to understand its importance.

We saw that as the solution to $u(r,t)$ blew up the solution to $v(r,t)$ remained the same. If we made changes to the initial data we may see a change in how these functions depend on one another.

Some of the solutions were not very smooth. We could use smoothing of the mesh to gain better approximations of the results, and return more meaningful results.

Bibliography

[1] **C.J. Budd, R. Carretero-Gonzalez, R.D. Russell**, Precise computations of chemotactic collapse using moving mesh methods, *Journal of Computational Physics*, **202**, 2005, 463-487.

[2] **C.J. Budd, Jianping Chen, W Huang, R.D. Russell**, Moving mesh methods with applications to blow-up problems for PDEs, 1995.

[3] **G.D.Smith:**, *Numerical Solution of Partial differential Equations: Finite difference Methods*, Oxford University Press, 1985.

[4] **T.Hillen, K.J.Painter**, A user's guide to PDE models for chemotaxis, *Journal of Mathematical Biology*, **58**, 2009, 183-217.

[5] **G.J. Barenblatt**, *Scaling*, Cambridge University Pres, 2003.

[6] **M.A. Herrero, J.J.L. Velazquez**, Singularity patterns in a chemotaxis model, *Math. Ann.*, **306**, 1996, 583-623.

[7] **W. Huang, Y. Ren, R.D. Russell**, Moving mesh methods based upon moving mesh partial differential equations, *SIAM Journal on Numerical Analysis*, **31**, 1994, 709-730.

[8] **W. Huang, R.D. Russell**, A moving collocation method for solving time dependent partial differential equations *Appl. Numer. Math.*, **20**, 1996, 101-116.

[9] **M.J. Tindall, P.K. Maini, S.L. Porter, J.P. Armitage**, Overview of Mathematical Approaches Used to Model bacterial Chemotaxis II: Bacterial Populations *Appl. Numer. Math.*, 2009.

[10] **E. Hilderband, U.B. Kaupp**, Sperm Chemotaxis: a primer, *Ann. N. Y. Acad. Sci.*, **1061**, 2005, 221-225.

[11] **J. Condeelis, R.H. Singer, J.E. Segall**, The great escape: when cancer cells hijack the genes for chemotaxis and motility, *Annu. Rev. Cell Dev. Biol.*, **21**, 2005, 695-718.

[12] **E.F. Keller, L.A. Segal**, Initiation of slime mold aggregation viewed as an instability, *J. Theor. Biol.*, **26**, 1970, 399-415.