Comparison of Semi-Implicit and Fully-Implicit Methods for a Highly Degenerate Diffusion-Reaction Equation Coupled with an ODE

by

Eric M. Jalbert

A Thesis
presented to
The University of Guelph

In partial fulfilment of requirements
for the degree of
Master of Science
in
Applied Mathematics

Guelph, Ontario, Canada

Contents

Chapter 1

Introductions

1.1 Background

- C. thermocellum biology and the connection to ethanol production. Look at USA data and the fact that C.thermo is a cellulolytic bacteria.
- Talk about exisiting models for biofilms, such as: suspended cultures, reator scale-models, and Cellular Automata.
- Look at PDE with Volume filling and compare the model we will use here to the traditional biofilm model.
- Describe the Numerics of PDE's, such as semi-implicit methods and the different work that has been done to solve these kinds of problems. Mention the problems those methods would have with our system and mention where this work fits in.

1.2 Objectives

This will be one to two paragraphs for each objective.

1. Numerical development, implementation and validation of a fully implicit method

2. The comparison fo the fully implciit method with the semi-implicit methods

3. Simulation work; see if we can better understand the mechanisms of the bacteria by extracting some interesting observations from our simulations.

1.3 Outline

This will be a short section that describes what each section after this one will be covering and how it helps to accomplish the objectives, either in bullet point or one sentence for each section.

When you write this section, also make sure that you describe how each of these sections/chapters relates to the objectives that you formulated, and how these sections relate to each other

Chapter 2

Model Definition

2.1 Model Description

The model used for simulations is based on the deterministic biofilm model developed in Eberl et al. (2001), which was designed for modelling the development of spatially heterogenous biofilm structures. They modelled the biomass density and nutrient concentration as a two-PDE-coupled system. Here the spatial diffusion of the nutrient concetration is removed to mimic the carbon fiberous substrate that *C.Thermocellum* consumes in growth. With that modification, a PDE-ODE-coupled system can be purposed as,

$$M_t = \nabla_x \left(d(M) \nabla_x M \right) + f(C) M \tag{2.1}$$

$$C_t = -g(C) M \tag{2.2}$$

where

$$d(M) = d \frac{M^\alpha}{(1-M)^\beta} \tag{2.3}$$

$$f(C) = u \frac{C}{k+C} - n \tag{2.4}$$

$$g(C) = y\frac{C}{k+C} \tag{2.5}$$

Here we have a pair of equations, (2.1) and (2.2), that represent the biomass density and substrate concentration respectivly. The spatial diffusion of the biofilm is modelled with density-dependent diffusion, represented by (2.3), and the growth rate of biomass is given by (2.4). The growth rate is simple Monod kinetic growth with a constant death rate. In (2.2) there is only a consumption term from the bacteria consuming the carbon substrate. This term is based on the growth of the biomass, differing only by a scalar multiplier.

The dimensions of the parameters and variables are in Tabel 2.1.

Variable/Parameter	Dimensions
t	$[days]$
x	$[meters]$
M	$[\frac{grams}{meters^3}]$
C	$[\frac{grams}{meters^3}]$
d	$[\frac{meters^2}{days}]$
α	$[-]$
β	$[-]$
u	$[days^{-1}]$
k	$[\frac{grams}{meters^3}]$
y	$[\frac{C}{M}]$
n	$[\frac{grams}{meters^3 \cdot days}]$

Table 2.1: List of parameters and their dimensions

2.2 Nondimensionalization

To help facilitate the analyses of this system, the full removal of all physical units is preferred. This process of nondimensionalization involves using known parameters to create substitutions with physical units cancelling. Here the parameters used are: the biomass growth rate, u; the length of the region, L; and the maximum density for biomass and substrate, M_∞ and C_∞. From using the following parameter changes, the system can be made unitless.

$$\chi = \frac{x}{L} \implies Ld\chi = dx \tag{2.6}$$

$$\tau = ut \implies \frac{1}{u}d\tau = dt \tag{2.7}$$

$$\mathcal{M} = \frac{M}{M_\infty} \tag{2.8}$$

$$\mathcal{C} = \frac{C}{C_\infty} \tag{2.9}$$

$$\delta = \frac{1}{uL^2}d \tag{2.10}$$

$$\kappa = \frac{k}{C_\infty} \tag{2.11}$$

$$\nu = \frac{n}{uC_\infty} \tag{2.12}$$

$$\gamma = \frac{M_\infty}{C_\infty}y \tag{2.13}$$

Using these, (2.1) and (2.2) can be simplified and nondimensionalized into,

$$\mathcal{M}_\tau = \nabla_\chi \left(D(\mathcal{M})\nabla_\chi \mathcal{M}\right) + F(\mathcal{C})\mathcal{M} \tag{2.14}$$

$$\mathcal{C}_\tau = -G(\mathcal{C})\mathcal{M}, \tag{2.15}$$

where,

$$\begin{aligned} D(\mathcal{M}) &= \delta \frac{\mathcal{M}^\alpha}{(1-\mathcal{M})^\beta} \\ F(\mathcal{C}) &= \frac{\mathcal{C}}{\kappa + \mathcal{C}} - \nu \\ G(\mathcal{C}) &= \gamma \frac{\mathcal{C}}{\kappa + \mathcal{C}}. \end{aligned} \tag{2.16}$$

with only $\delta, \kappa, \nu, \gamma$ as model parameters.

2.3 Parameters

Each of the dimensionless parameters in (2.16) have a biological representation based on the transformations done. The parameter δ is the dimensionaless constant for diffusion. It affects the change in biomass from adjacent biomass sources, a greater δ results in a greater change. The parameter κ is the half-saturation point, it is exactly the value for which substrate concentration results in 0.5-optimum growth rate. Parameter ν is the death rate of the biomass. Specifically, it is the ratio of biomass growth to death, representing the fraction of biomass density that perishs from natural causes or a lack of substrate. Lastly, γ is the yield ratio. It signifies the ratio of substrate consumed to biomass growth. Here, a larger γ value results in more substrate being consumed to produce the same amount of biomass.

With (2.14) being reduced to four parameters the numerical analysis become more simiplified while still retaining the same significance in results.

Chapter 3

Numerical Methods

3.1 Discretization

In order to find the solution for (2.14) spatial and temporal discretizations must be made. First the equations are discretizied in time,

$$\frac{M^{k+1} - M^k}{\Delta t} = \nabla_x (D(M^{k+1}) \nabla_x M^{k+1}) + F(C^{k+1}) M^{k+1}, \tag{3.1}$$

$$\frac{C^{k+1} - C^k}{\Delta t} = \frac{h}{2} (G(C^{k+1}) M^{k+1} + G(C^k) M^k). \tag{3.2}$$

Here, (3.1) follows the ideas of the Backwards Euler Method; (3.2) follows Trapezoidal Rule. The index variable k has also been introduced in (3.1 - 3.2) such that $M^k(x) \approx M(t^k, x)$, allowing an approximation at a certain time, t^k, to be used; this reduces the dimensionality of the problem.

For this system, the region of consideration will be a rectangular region, Ω. This region has Neumann boundary conditions, $\frac{\partial M}{\partial x} = \frac{\partial C}{\partial x} = 0, \forall x \in \partial\Omega$. Now, only (3.1) requires spatial considerations since, according to the biology of our system, the substrate does not diffuse across the region. The spatial discretization will be through the Finite Difference Method as described in Saad (2003). Here, a uniform $n \times m$ grid is used to discretize Ω. Since all the calculations will be done on the grid intersections the discretization will be grid-point based. This means that a $n \times m$ grid implies there

are $(n-1) \times (m-1)$ grid boxes. The distance between grid points is the same in both x_1 and x_2 dimensions; we have $\Delta x_1 = \Delta x_2$. A five-point stencil is used to approximate the solution of (3.1) at each grid point. To index the grid point, i and j are used such that $M^k_{i,j} \approx M(t^k, x_{1_i}, x_{2_j})$. To account for the dependency on neighbouring grid points, we introduce σ as the index pair from the set

$$\mathcal{N}_{ij} = \{n_{ij}, e_{ij}, s_{ij}, w_{ij}\}. \tag{3.3}$$

where,

$$\begin{aligned} n_{ij} &= \begin{cases} (i, j+1) & \text{if } j < m \\ (i, j-1) & \text{if } j = m \end{cases} \qquad e_{ij} = \begin{cases} (i+1, j) & \text{if } i < n \\ (i-1, j) & \text{if } i = n \end{cases} \\ s_{ij} &= \begin{cases} (i, j-1) & \text{if } j > 0 \\ (i, j+1) & \text{if } j = 0 \end{cases} \qquad w_{ij} = \begin{cases} (i-1, j) & \text{if } i > 0 \\ (i+1, j) & \text{if } i = 0 \end{cases}. \end{aligned} \tag{3.4}$$

With $\mathcal{N}_{ij}$ and σ we can account for the difference in boundary points and interior points.

The equation for (3.1), after spatial discretization, is

$$\frac{M^{k+1}_{i,j} - M^k_{i,j}}{\Delta t} = \frac{1}{\Delta x^2} \sum_{\sigma \in \mathcal{N}_{ij}} \left(\frac{D(M^{k+1}_\sigma) + D(M^{k+1}_{i,j})}{2} \right) \cdot \left(M^{k+1}_\sigma - M^{k+1}_{i,j} \right) + F(C^{k+1}_{i,j}) M^{k+1}_{i,j} \tag{3.5}$$

For (3.5), the arithmetic mean of the diffusion function, D, is taken because of the steep gradiant at the interface. The alternative would be to use $D(M^{k+1}_{i+\frac{s}{2}, j+\frac{r}{2}})$, however this may result in a value of zero and thus nullify the effect of the spatial diffusion.

3.2 Solving Technique

Now there exist equations for which C and M can be solved, (3.2) and (3.5) respectivly. Using C^k and $M^k_{i,j}$ as approximations of the solutions for (2.14) will allow the system to be solved by computing C^{k+1} and $M^{k+1}_{i,j}$. However, there are complications with trying to get an explicit formula for $M^{k+1}_{i,j}$ from (3.5) because of the dependency on M in $D(M)$. To remedy this, a fixed point iteration is

introduced. In a single time step, the solutions for M and C can be solved using the previous time step solution in the follow manner:

$$\frac{M_{i,j}^{(p+1)} - M_{i,j}^{k}}{\Delta t} = \frac{1}{\Delta x^2} \sum_{(s,r)\in\mathbb{A}} \left(\frac{D(M_{i+s,j+r}^{(p)}) + D(M_{i,j}^{(p)})}{2} \cdot (M_{i+s,j+r}^{(p+1)} - M_{i,j}^{(p+1)}) \right) + F(C_{i,j}^{(p)}) M_{i,j}^{(p+1)} \tag{3.6}$$

$$\frac{C^{(p+1)} - C^{k}}{\Delta t} = \frac{-1}{2}(G(C^{(p+1)})M^{(p+1)} + G(C^{k})M^{k}) \tag{3.7}$$

where $(p) \in (0, 1, \ldots, P)$. Note, that the equation for $M_{i,j}^{(p+1)}$ shown in (3.6) refers to the interior points only. A similar change is done for the boundary points but is not shown due to its complexity. It is important to show explicitly that the purpose of the fixed point iteration is to link two distinct times with P solutions in between them, such that:

$$\begin{aligned} M^{(p=0)} &= M^{k}, \quad M^{(p=P)} = M^{k+1}, \\ C^{(p=0)} &= C^{k}, \quad C^{(p=P)} = C^{k+1}. \end{aligned} \tag{3.8}$$

In this fixed point format, given by (3.6 - 3.7), the equations can be rearrange and solved by conventional methods.

For (3.6), a linear system of equations can be created following Saad (2003). For each grid point (i, j) a linear system exists, defined as:

$$\begin{aligned} \frac{M_{i,j}^{k}}{\Delta t} = &\sum_{(i,j)\in\mathbb{A}} \left(\frac{D(M_{i+s,j+r}^{(p+1)}) + D(M_{i,j}^{(p+1)})}{2\Delta x^2} \cdot M_{i+s,j+r}^{(p+1)} \right) \\ &+ \left(\sum_{(i,j)\in\mathbb{A}} \left(\frac{D(M_{i+s,j+r}^{(p+1)}) + D(M_{i,j}^{(p+1)})}{2\Delta x^2} \right) - F(C_{i,j}^{(p)}) + \frac{1}{\Delta t} \right) M_{i,j}^{(p+1)}. \end{aligned} \tag{3.9}$$

From (3.9), a five-diagonal matrix can be created defined as,

$$A = \begin{pmatrix} a_{i,j} & a_{i+1,j} & & a_{i,j+1} & & & & & \\ a_{i-1,j} & \ddots & \ddots & & \ddots & & & & \\ & \ddots & \ddots & \ddots & & \ddots & & & \\ a_{i,j-1} & & a_{i-1,j} & a_{i,j} & a_{i+1,j} & & a_{i,j+1} & & \\ & \ddots & & \ddots & \ddots & \ddots & & \ddots & \\ & & a_{i,j-1} & & a_{i-1,j} & a_{i,j} & a_{i+1,j} & & a_{i,j+1} \\ & & & \ddots & & \ddots & \ddots & \ddots & \\ & & & & \ddots & & \ddots & \ddots & a_{i+1,j} \\ & & & & & a_{i,j-1} & & a_{i-1,j} & a_{i,j} \end{pmatrix} \tag{3.10}$$

where each $a_{i,j}$ is the coefficient based on (3.9).

Solving (3.10) can be done by use of the Conjugate Gradient method provided that certain conditions are satisfied.

Proposition 3.2.1. *The matrix A, (3.10), is positive definite and symmetric when $\frac{1}{F(C_{i,j}^{(p)})} < \Delta t$.*

Proof. Matrix A is positive definite if all the eigenvalues are positive. Using the Circle theorem described by Geršgorin (1931), the eigenvalues can be shown to be positive if, independently on all rows, the sum of the off-diagonals values is less then the diagonal value. This can be verified. From (3.9) it can be said that,

$$\sum_{(i,j)\in\mathbb{A}} \left(\frac{D(M^{(p)}_{i+\frac{s}{2},j+\frac{r}{2}})}{\Delta x^2} \right) < \left(\sum_{(i,j)\in\mathbb{A}} \left(\frac{D(M^{(p)}_{i+\frac{s}{2},j+\frac{r}{2}})}{\Delta x^2} \right) - F(C_{i,j}^{(p)}) + \frac{1}{\Delta t} \right). \tag{3.11}$$

This simplifies to,

$$F(C_{i,j}^{(p)}) < \frac{1}{\Delta t} \tag{3.12}$$

The symmetry of A can be trivially shown if one considers the formation of the diagonals. On a

single row, each element corresponds to the adjacent grid points of grid i, j. As the grid ordering counts along, the elements that are equidistance from the diagonal are actually reference to the same grid point. Therefore we have symmetry. □

It is important to remark that even though there is a condition for which matrix A is positive definite and symmetric, it realistically will never occur. The condition, $\frac{1}{F(C)} < \Delta t$, relates the growth of the biomass to the size of timestep selected. Specifically, if a large enough time step is choosen, then A is not guarenteed to converge. When this occurs, it means that a time step, larger then the characteristic growth rate of the biomass, has been incorrectly choosen. This means that the there would be no relavent results since all the growth, and subsequent reactions, would have occured in a single timestep.

Given that A is positive definite and symmetric, the conjugate gradiant method can be used to compute the solution. As an added property, A also happens to be diagonally dominate. This results in A being a M-matrix. It also means that it could be solved using Bi-Conjugate Gradient Method. However the Conjugate Gradient method has a faster computation time then Bi-Conjugate Gradiant method for this problem and is used for this reason (Barrett et al. (1987)).

For solving (3.7), the equation can be rearranged into a quadratic form, substituting in $G(C)$ from (2.16)

$$\left(C^{(p+1)}\right)^2 + \left(\kappa - C^k + \frac{h}{2}\gamma M^{(p+1)} + \frac{h}{2}\frac{\gamma C^k M^k}{\kappa + C^k}\right) C^{(p+1)} + \left(-\kappa C^k + \frac{h}{2}\frac{\gamma \kappa C^k M^k}{\kappa + C^k}\right) = 0. \quad (3.13)$$

Using the quadratic equation results in,

$$C^{(p+1)} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (3.14)$$

for which,

$$\begin{aligned} a &= 1 \\ b &= \kappa - C^k + \frac{h}{2}\gamma M^{(p+1)} + \frac{h}{2}\frac{\gamma C^k M^k}{\kappa + C^k} \\ c &= -\kappa C^k + \frac{h}{2}\frac{\gamma \kappa C^k M^k}{\kappa + C^k} \end{aligned} \tag{3.15}$$

Unless $b^2 - 4ac = 0$, we have two different solutions to $C^{(p+1)}$. The problem with that is that if both solutions are positive we have two valid values to be used. If the quantitative behaviour of the solutions change with choices of timestep, h, then the validity of the method goes down. Here, we can show that there will always be only one positive solution, regardless of timestep choice.

Proposition 3.2.2. *The quadratic equation defined as (3.13) will always have one positive solution and one negative solution for realistic parameter choices.*

Proof. Rearranging (3.13) so that all the h terms are on the right-hand-side, we get

$$\left(C^{(p+1)}\right)^2 + \left(\kappa - C^k\right) C^{(p+1)} - \kappa C^k = \left(\frac{\gamma C^k M^k}{2\left(\kappa + C^k\right)} - \left(\frac{\gamma M^{(p+1)}}{2} - \frac{\gamma C^k M^k}{2\left(\kappa + C^k\right)}\right) C^{(p+1)}\right) h. \tag{3.16}$$

To simplify analysis, we let $\bar{a} = \frac{\gamma M^{(p+1)}}{2} - \frac{\gamma C^k M^k}{2\left(\kappa + C^k\right)}$ and $\bar{b} = \frac{\gamma C^k M^k}{2\left(\kappa + C^k\right)}$.

We analysize both the left-hand-side and right-hand-side independently by letting $f_l = \left(C^{(p+1)}\right)^2 + \left(\kappa - C^k\right) C^{(p+1)} - \kappa C^k$ and $f_r = \left(\bar{b} - \bar{a} C^{(p+1)}\right) h$. f_l is a quadratic equation with positive concavity everywhere and $C^{(p+1)}$-intercept at $-\kappa C^k < 0$. f_r is a line with a slope opposite to the sign of $\bar{a}$ and has $C^{(p+1)}$-intercept at $\bar{b}h > 0$. Since both f_l and f_r are defined for all $\mathcal{R}$ they must intersect at some point. □

Now that computable solutions for M and C at a single time step have been found, an algorithm to solve for the next time step can be established. Algorithm 1 shows the organization of solving (3.7 - 3.6).

Note that Algorithm 1 actually describes both a fully- and semi- implicit method for solving (2.14).

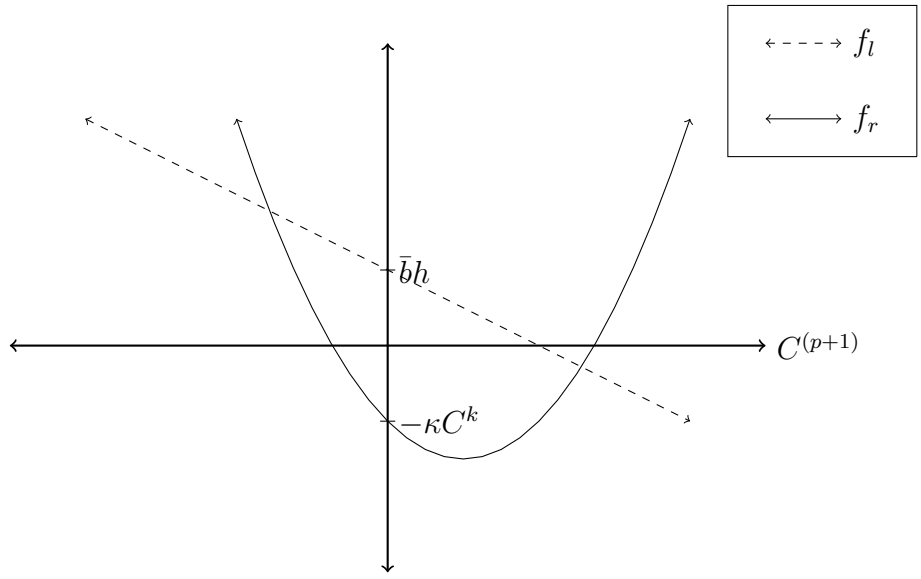


Figure 3.1: Graph of $f_l = \left(C^{(p+1)}\right)^2 + \left(\kappa - C^k\right) C^{(p+1)} - \kappa C^k$ and $f_r = \left(\bar{b} - \bar{a}C^{(p+1)}\right) h$ for $\bar{a} > 0$ and $\kappa - C^k < 0$. Notice that because $-\kappa C^k < 0$ and $\bar{b}h > 0$ for all realistic parameter values the two functions will always intersect in the positive $C^{(p+1)}$ region. Even if $\bar{a} < 0$ or $\kappa - C^k > 0$, the functions are guaranteed to have only one positive intersection since the location of the y-intercepts will not qualitativly change.

Data: M^k, C^k are the values from the previous timestep and $p = 0$.
begin
  Let $M^{(0)} = M^k$ and $C^{(0)} = C^k$;
  while *Convergence is not acheived* **do**
    Solve $A^{(p)} M^{(p+1)} = b^{(p)}$;
    Solve $C^{(p+1)} = \frac{1}{2}\left(2b \pm \sqrt{b^2 - 4c}\right)$;
    Check convergence;
    Let $C^{(p)} = C_{(p+1)}$;
    Let $M^{(p)} = M_{(p+1)}$;
    Let $p = p + 1$;
  end
  Let $M^{(k+1)} = M^{(P)}$ and $C^{(k+1)} = C^{(P)}$;
end

Algorithm 1: Algorithm for the fully-implicit solving of (2.14)

If $P = 1$ then only a single iteration of the algorithm is applied, which correlates to a semi-implicit method would behave. This would result in a change similar to how the Gauss-Seidal method changes the Jacobi method; the values used would no longer be updated in a single timestep when $P = 1$.

To use the algorithm, the matrix system was converted into a 1D array by use of a bijective mapping defined as:

$$\begin{array}{rccc} \pi: & \{0,\ldots,n\}\times\{0,\ldots,m\} & \rightarrow & \{0,\ldots,nm\} \\ & (i,j) & \rightarrow & \pi(i,j) \end{array} \tag{3.17}$$

This mapping allows the system to be easily stored in diagonal format, since (3.10) has five distinct diagonals.

3.3 Computational Setup

The implementation of Algorithm 1 was done with Fortran.

All the computations were run on a custom built workstation with an Intel Xeon CPU E5-2650 (1.2 GHz, 20MB cache size) and 32 GB RAM under Red Hat Enterprise Linux Server release 6.5 (Santiago). Running the computations with OpenMP, took advantage of 6 out of the 16 processors of the Intel Xeon CPU, each with 2 threads. The GNU Fortran compiler, version 4.4.7, was used for all computations; the compiler arguments were

```
-O3 -fdefault-real-8 -fopenmp
```

3.4 Method Validation

With a defined method and computational setup a varaity of simulations can be run to observe the accuracy and behaviour of the method. An examination of a typical simulation will show if the expected behaviour is observed, validating the method as functioning. A convergence analysis for the method can be done to confirm that solutions from different grid sizes approach a single solution as they become more precise. This convergence test will also show the thresholds for an accurate simu-

lation result, to help reduce the computiation times. With a well-established method, the comparison between semi- and fully-implicit methods can be done.

subsectionBasic Simulations

Using Algorithm 1, simple scenarios can be tested as a first verification on the method.

A simple test would be to check if the spatial discretization can preserve specific characteristics of the solutions. One example of this would be seeing if a 1D initial condition could be preserved as time progresses. Having all of the biomass on one boundary of Ω, for example across the y-axis, would qualify as a 1D intial condition. These initial conditions will be defined as:

$$M = \begin{cases} -\left(\frac{h}{d^4}\right) x^4 + h & \text{, if } y \leq d \\ 0 & \text{, otherwise} \end{cases} \tag{3.18}$$
$$C = 1$$

where $h = 0.1$ and $d = \frac{5}{128}$. Here, h and d represent the height and depth of the innoculation site.

The solution shown in Figure 3.2 shows that the 1D characteristic of the biomass stays at a later time.

Another characteristic to observe would be if a spherical initial condition remains spherical. Using intial conditions for the biomass,

$$M = \begin{cases} -\frac{h}{d^2}\left((x-0.5)^2 + (y-0.5)^2\right) + h & \text{, if } (x-0.5)^2 + (y-0.5)^2 < d^2 \\ 0 & \text{, otherwise} \end{cases}, \tag{3.19}$$

a test can be tried to see if the spherical nature of the solution is kept as time progresses. This can be seen in Figure 3.3, where at different times the shape of the biomass, M, is seen to remain spherical.

Both Figure 3.2 and Figure 3.3 increase the confidence that the spatial discretization did not introduce any loss of characteristics for the solutions.

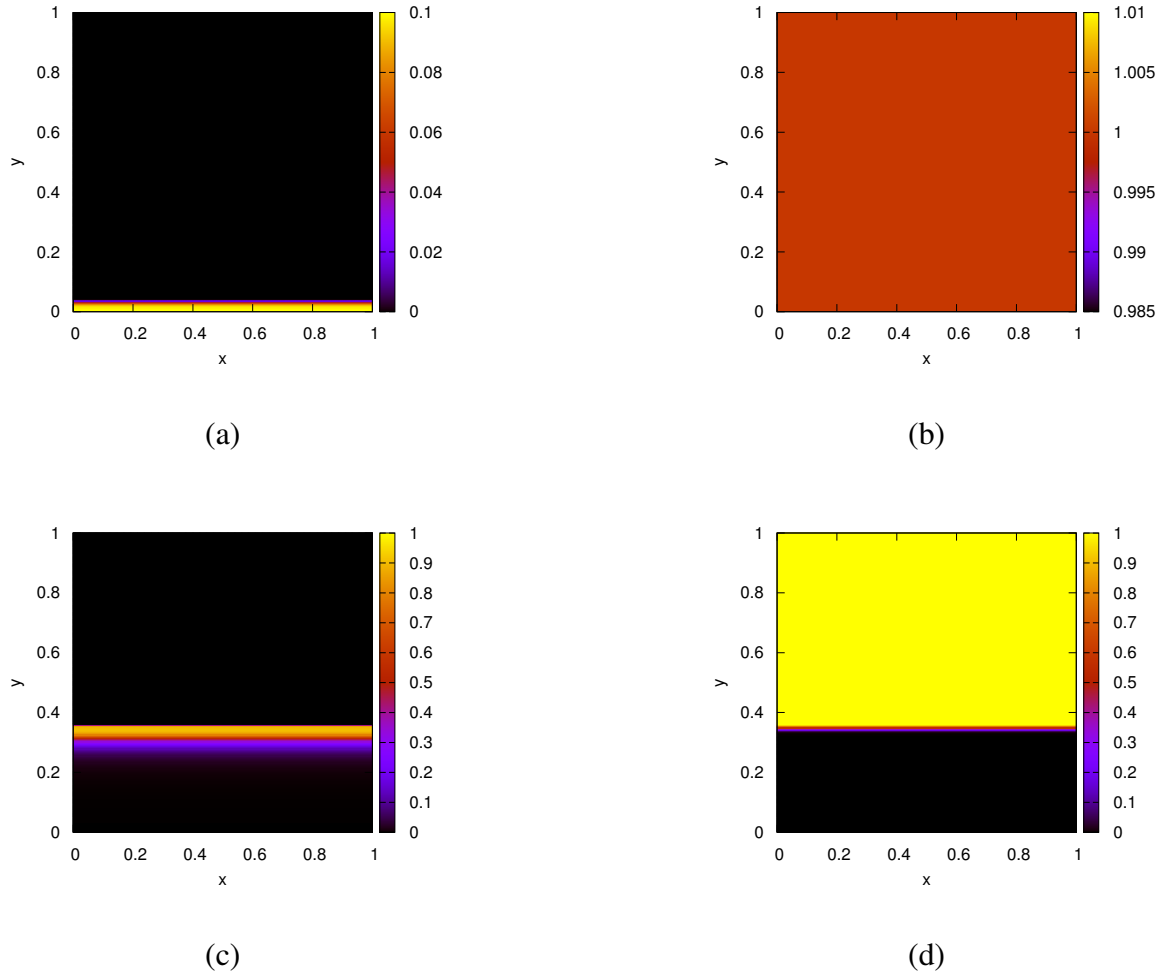
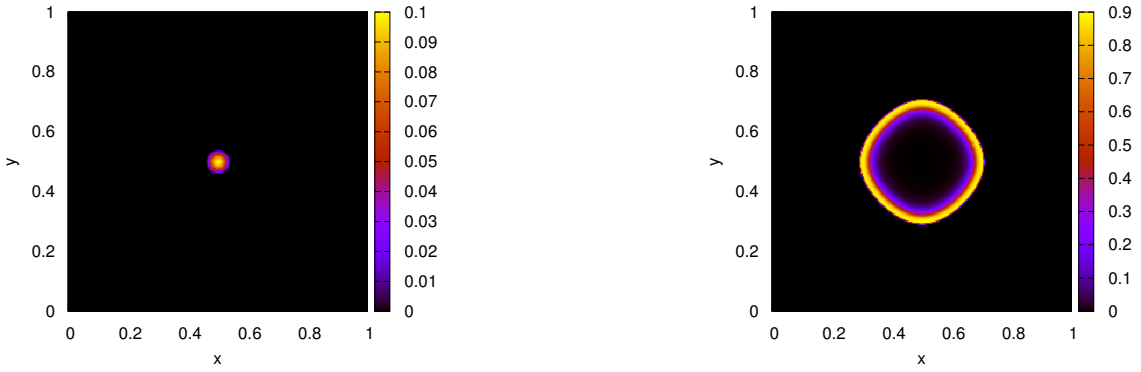


Figure 3.2: Solutions for (ac) M and (bd) C with 1D initial conditions defined in (3.18) at (ab) $t = 0$ and (cd) $t = 40$. Computed with a 1025×1025 grid and a timestep of $\Delta t = 10^{-3}$.



(a) (b)

Figure 3.3: Solutions for M with spherical initial conditions defined by (3.19) at (a) $t = 0$ and (b) $t = 40$. Computed with a 1025×1025 grid and a timestep of $\Delta t = 10^{-3}$.

Given the boundary conditions and spatial discretization, there could be a possible source or sink of biomass when it must diffuse along the boundary of the region. To ensure this is not the case, the total amount of biomass can be used to compare the simulated amount against the theoretical amount. However, the total biomass cannot be exactly determined with the given growth rate function. This means that there will not be anything to measure the validity of the simulation solution against. If we let the growth rate be some constant called a, the exact total biomass can be calculated.

the expected total biomass would be of the form $y_0 e^{at}$. This can be checked by tracking the total biomass, now called $T_M(t)$, with the changed growth rate function, $F(C) = a$. The calculation of $T_M(t)$ can be done by,

$$T_M(t) = \int_\Omega M(t) dx. \tag{3.20}$$

Numerically, this is computed by grid-wise summation,

$$T_M(t^k) \approx T_M^k = \frac{\sum_i^n \sum_j^m M_{i,j}^k}{nm}. \tag{3.21}$$

The simulation setup used will be analogous to that used for Figure 3.3. The one difference will be that the simulation here is ran for a longer time to allow the biomass to diffuse along the boundary,

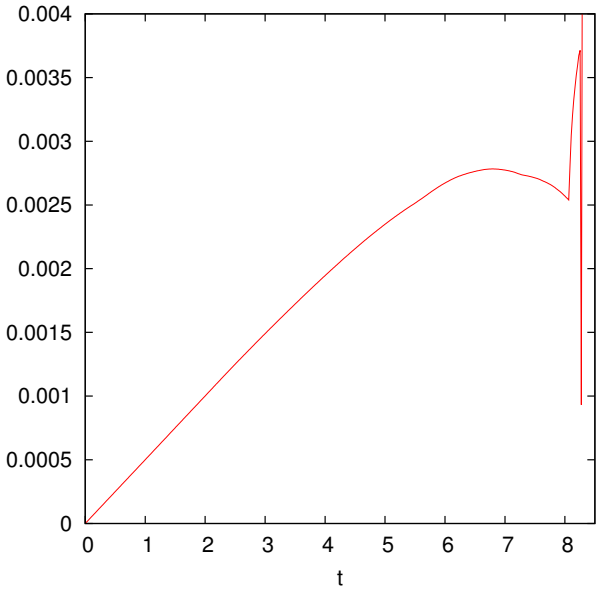


Figure 3.4: Plot of the relative error, $\frac{|f_1-f_2|}{|f_2|}$, between the computed total biomass, $f_1 = T_M(t)$, and the theoretical total biomass, $f_2 = y_0 e^x$. The changes after $t = 8$ are from the biomass having completely filled the region Ω. This means that there is no physical space for the biomass to occupy and thus the growth slows down to a stop.

showing the boundary effects.

From Figure 3.4 we can see that the total biomass only differs between the computed value and the theoretical value by a relative error less then 0.003. The cases where the error becomes significant are from the region being completely filled with biomass, at which point diffusion is no longer possible. The error fluctuates violently here because of this. This suggests that the method does not introduce any significant sources or sinks of biomass at the boundary of the region.

subsectionConvergence Analysis To validate the accuracy of the method, convergence analyses on the spatial discretizations will need to be made. Then the comparison between the semi- and fully-implicit method established in Algorithm 1 can investigated. First, a metric must be formed to enable consistent comparisons between different simulation solutions. This metric will be referred to as the error.

subsubsectionError Computations

The error is computed by taking the relative normed-difference between two solution in the following

fashion:

$$\epsilon_{sol} = \frac{||u_1 - u_2||}{||u_2||} \tag{3.22}$$

where u_1 represents one simulation solution and u_2 represents the solution that is theoretically more accurate. The theoretical accuracy of u_2 derives from the fact that most comparisons will be done between solutions where one is trivially expected to be more precise. For our purposes, the solutions we compare will typically vary in only Δx or between semi- and fully- implicit. These are understood to have the relation that a smaller Δx, and that the fully-implicit method with the highest tolerance is to be more accurate. There is an assumption that both u_1 and u_2 have the same number of grid points, so that the difference can be taken grid-wise.

The results of the error computations, named ϵ_{sol}, is a numerical value for the difference between two solutions. This depends on the norm used during the computations. Here three norms will be used:

$$\ell_1 : ||u||_1 = \frac{1}{nm} \sum_{\pi(i,j)}^{nm} |u_{i,j}| \tag{3.23}$$

$$\ell_2 : ||u||_2 = \frac{1}{nm} \sqrt{\sum_{\pi(i,j)}^{nm} (u_{i,j})^2} \tag{3.24}$$

$$\ell_\infty : ||u||_\infty = \max_{\substack{i=1,\dots,n \\ j=1,\dots,m}} |u_{i,j}| \tag{3.25}$$

These different norms will all be used to create a broader understanding of the error. This creates three distinct values for ϵ_{sol}, named ϵ_{ℓ_1}, ϵ_{ℓ_2}, and ϵ_{ℓ_∞}; each named for the norm used during the computation.

subsubsectionGrid Size Convergence To observe the validity of the method, a test on the convergence of solutions based on the spatial discritization is done. This will involve using the same simulation described in (3.18) due to the simplicity.

The convergence will be tracked with only two forms of ϵ_{sol}; ϵ_1 and ϵ_2. This is because the value of ϵ_∞ doesn't vary with the grid size, since the wave front has a steep interface and tends to lead to inconsistent changes in error. Since the use of ϵ_∞ is not a suitable method for measuring the error, the

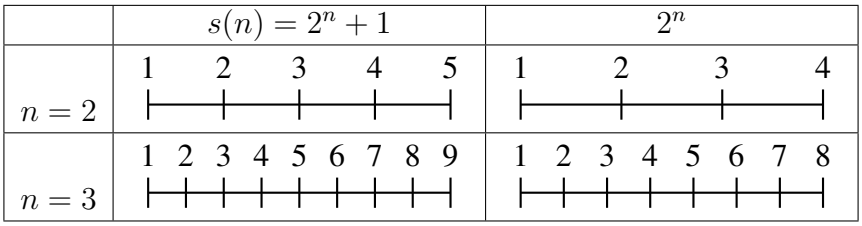


Figure 3.5: Visualization in 1D to illustrate the choice of $s(n) = 2^n + 1$ instead of 2^n for the grid size selection. Here is can be seen that successive grid size selections using $s(n)$ line up on certain grid points and when using 2^n no grid points are equivalent.

inconsistency does not suggest an invalidity with the method. Because of the difference in the number of grid points between different solutions, u_1 and u_2, only the grid points in the coarser refinement will be used. This places a limitation on the selection of grid-sizes since there must be some grid points locations that are the same for two different chosen grid sizes. For this purpose, we define the function $s(n) = 2^n + 1$ for $n \in \mathbb{N}$ to be used as the grid size selection function. Now certain grid points will match without the use of linear interpolation. Here we do not use the typical grid doubling, 2^n, because no grid points equal between two different grid size selection, as illustrated in Figure 3.5.

Using the same simulation setup as was done in Figure (3.2), solutions resulting from different grid sizes based on $s(n)$ are computed for $n = 5, 6, \ldots, 12$. In this case, when calculating $\epsilon_{sol} = \frac{|u_1 - u_2|}{|u_2|}$, we let u_1 be the grid size under investigation and u_2 be the solutions of the most refined gridsize, $n = 12$. This results in a series of values that, if the solutions converge with smaller grid sizes, should have less change.

The results from Figure 3.6 show that the solutions converge as the grid size become refined with the grid size.

3.5 Comparison of Semi-implicit and Fully-implicit Method

Here the main comparison that analyses the effects of using Algorithm 1 with different tolerances. Recall that the main observation is for $tol. = 1$, which correlates to the semi-implicit method since it will allow only a single iteration of the algorithm.

The simulation used is the same as described in (3.2). The comparison will be on multiple metrics:

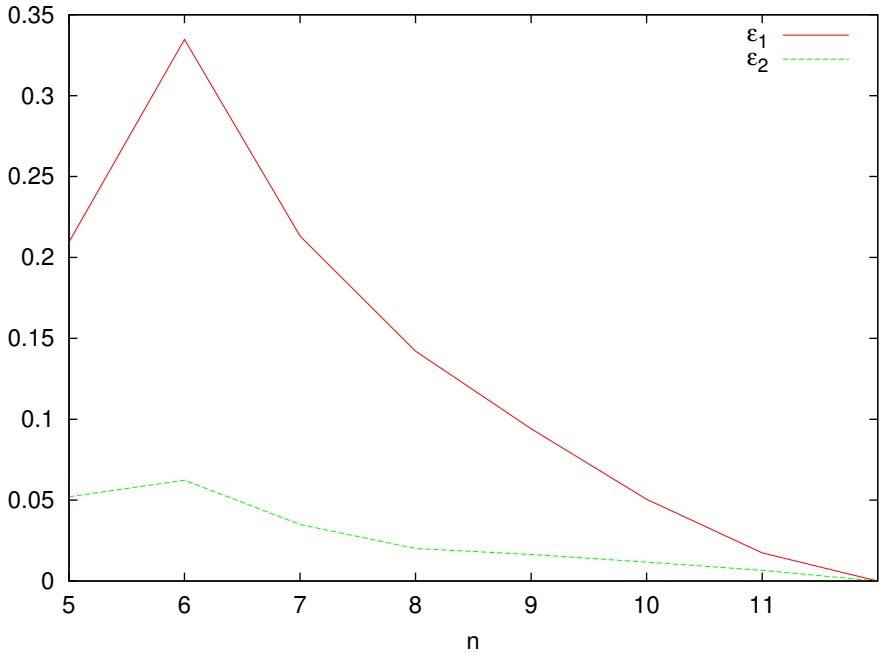


Figure 3.6: Plot showing the convergence of solutions based on changes in Δx. The computions are of ϵ_{ℓ_1} and ϵ_{ℓ_2} with grid-size following $s(n) = 2^n + 1$.

the average number of iterations of Algorithm 1, the value of ϵ_1 and ϵ_2, the computation time of the simulation, and the location of the wave peak.

The average number of iterations are tracked so that an idea of the extra work can be formed. As the tolerance decreases the amount of iterations the algorithm must perform will increase, the degree of increase will help relate the amount of work.

The value of ϵ_1 and ϵ_2 act as a measure of accuracy. Here, these values correspond to the difference between a pair of solutions, u_1 and u_2. The (u_1, u_2) pairs are: $(1, 10^{-8}), (10^{-8}, 10^{-10}), (10^{-10}, 10^{-12}), (10^{-12}, 10^{14})$. Each row of Table 3.1 refers to the u_1 values of the pairs. Each difference was taken at the last timestep.

Along with accuracy, the simulation time is tracked. This is because it represents another metric for which the viability of the fully-implicit method can be verified. Theoretically there should be a decrease in the error with the fully-implicit method as the value for tol decreases. Therefore, this needs to be weighted against the cost of computational intensity and the increase of the simulation time.

The location of the wave peak is a tracked quality of the solution that reveals how consistent the results are. The wave peak is described here as the maximum value of the solution at the final timestep calculated. The ultimate goal is that the simulation solutions be converging towards the exact solution. To see this here the x-coordinate of the wave peak is tracked.

The results of the method comparison can be seen in Table 3.1.

Tol.	Avg. Iter.	ϵ_1	ϵ_2	Time	Wave Peak
1.0e-0	1.00000000	0	0	12.183000000	0.46484375
i.0e-1	1.00000000	0	0	12.208000000	0.46484375
1.0e-2	1.00000000	0	0	12.337999999	0.46484375
1.0e-3	1.00000000	0	0	12.231000000	0.46484375
1.0e-4	1.00000000	0.00200018013	0.000754518705	12.320000000	0.46484375
1.0e-5	1.96505087	2.44746455e-07	1.62832139e-07	18.986999998	0.4609375
1.0e-6	2.00000000	2.81448171e-07	1.94903038e-07	19.091000001	0.4609375
1.0e-7	2.00187495	1.12595285e-05	7.72252591e-06	19.094000001	0.4609375
1.0e-8	2.58568535	7.16665603e-07	1.74044990e-07	20.516999999	0.4609375
1.0e-9	2.90125246	1.05877879e-05	7.00561295e-06	21.308000000	0.4609375
1.0e-10	3.2278443	0.000143911078	9.66349067e-05	22.228000002	0.4609375
1.0e-11	16.099022	4.02028944e-05	2.71613115e-05	57.558999997	0.4609375
1.0e-12	36.318492	4.23049177e-06	2.86249772e-06	113.99400000	0.4609375
1.0e-13	57.681232			173.84899999999999	0.460937500000

Table 3.1: Results from running simulations with different Tol.

Chapter 4

Simulation Results

4.1 Typical Simulation

THe main point here is to show the results, visually, of a typical simulation. This means there will be a number of points to discuss here:

- Describe the initial conditions and region
- Give a verbal description of what the simulation will be.
- Show the parameter values
- Show the results.

4.2 Travelling Wave Analysis

This will be the beefy section that details the travelling wave analysis. Some things to include here

- show that the 2d-region problem can be reduced to a 1d-problem with appropriatly homogenous initial conditions in one of the dimensions (x2?)
- Show the typical solution of the travelling wave

- Show computations for the wave speed
- show the results of the wavespeed as a function of $\mu, \delta, \nu, \gamma, \kappa$ in a panel plot sort of thing.

4.3 Spatial Effects

This will be a quick section that goes through:

- A quick blerb on what the spatial effects could be and what they can effect. The idea here is that if there are no spatial effects then you can efficently ignore spatial terms and further simplify the model in the future?
- Go through the test of showing how two IC that differ spatially (clumped in a corner vs. uniform distribution on one side) talk about the results.

Chapter 5

Conclusions

5.1 Lessons Learned

This is be a series of bullet point paragraphs with each bullet having a lesson learned.

- So the idea is for each paragraph to have 3-4 sentences that open with main idea of the lessson. Then it can go into what the actual lesson was. Followed by what result gave this lesson. And then finish up with how this lesson will actually help with things.
- There should be a lesson for each of the main ideas discussed in this thesis. So for the numerics there should be a lesson on the comparison of semi- and fully-implicit methods. For the simulation there should be a number of results.

5.2 Future Work

This will be a quick section. Mainly have bullet point paragraphs again (like in ”lessons learned”) Each bullet paragraph will have:

- What could have been changed/improved/explored/avoided?
- How could the change be made?

- What could this gain?
- What possible challenges could this change make?

The focus of this section should be for the *non-trivial* points. That is unless it is too difficult to find good points.

References

R. Barrett, M. Berry, Chan T.F., J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Society for Industrial and Applied Mathematics, 1st edition, 1987.

J.C. Butcher. *Numerical Methods for Ordinary Differential Equations*. Wiley, 2nd edition, June 2008.

A Dumitrache. *Understanding Biofilms of Anaerobic, Thermophilic and Cellulolytic Bacteria: A Study towards the Advancement of Consolidated Bioprocessing Strategies*. PhD thesis, University of Toronto, 2014.

HJ Eberl and L Demaret. A finite difference scheme for a doubly degenerate diffusionreaction equation arising in microbial ecology. *Electron. J. Differential Equations*, page 7795, 2007.

HJ Eberl, DF Parker, and MCM van Loosdrecht. A new deterministic spatio-temporal continuum model for biofilm development. *Journal of Theoretical Medicine*, pages 161–175, 2001.

S. Geršgorin. über die abgrenzung der eigenwerte einer matrix. *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na*, pages 749–754, 1931.

DR Noguera, G Pizarro, DA Stahl, and BE Rittmann. Simulation of multispecies biofilm development in three dimensions. *Water Science and Technology*, 39:123–130, 1999.

BE Rittmann and PL McCarty. Model of steady-state biofilm kinetics. *Biotechnology and Bioengineering*, 22:2343–2357, 1980.

Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematic, 2nd edition, 2003.

S Sirca and M Horvat. *Computational Methods for Physicistsl: Compendium for Students*. Springer, 2012.

Z-W Wang, S-H Lee, JG Elkins, and JL Morrell-Falvey. Spatial and temporal dynamics of cellulose degradation and biofilm formation by caldicellulosiruptor obsidiansis and clostridium thermocellum. *AMB Express*, 1:30, 2011.

406