Numerical Analysis of Methods for Simulating Clostridium Thermocellum

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

List of Figures

List of Tables

Chapter 1

Model Definition

1.1 History

The tradition biofilm model has been continually developed over many iterations since 1980. Rittmann and McCarty (1980) formulated the steady-state biofilm model, developed using the concept that biofilm growth would be the result of a steady flux from substrate. Since then the model have evolved to include modelling three-dimensional growth of multispecies anaerobic biofilms (Noguera et al. (1999)) and spatially heterogeneous biofilm structures (Eberl et al. (2001)).

The modelling of *Clostridium Thermocellum* is unique because this celluloytic anaerobic bacteria does not generate an extracellular polymeric substance. This uncharacteristic behaviour means that the mathematical model based on the work of Eberl and Demaret (2007) cannot be used as is. They modelled the biomass density and nutrient concentrations as a two-PDE-coupled system. Recently, Wang et al. (2011), used a cellular automata based model for simulating the growth of *Clostridium Thermocellum*. From this, better results were thought to derive from a continuous differential equation based model. Here the spatial diffusion of the substrait concentration is removed to mimic the carbon substrait that is consumed by *Clostridium Thermocellum*. This results in a PDE-ODE-coupled system. This is based on the work done by Dumitrache (2014), where this same coupling was used and formulated.

1.2 Model Description

The model used for simulations is based on the deterministic model developed in Eberl et al. (2001), which was designed to simulate 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 substrait that *C.Thermocellum* consumes in growth. This makes a PDE-ODE-coupled system purposed as,

$$M = \nabla_x \left(d(M)\nabla_x M\right) + f(C)M \tag{1.1}$$

$$C = -g(C)M \tag{1.2}$$

where

$$d(M) = d\frac{M^\alpha}{(1-M)^\beta} \tag{1.3}$$

$$f(C) = u\frac{C}{k+C} - n \tag{1.4}$$

$$g(C) = y\frac{C}{k+C} \tag{1.5}$$

Here we have a pair of equations, (1.1) and (1.2), that represent the biomass density and substrait concentration respectivly. The spatial diffusion of the biofilm is modelled with density-dependent diffusion, represented by (1.3), and the growth rate of biomass is given by (1.4). The growth rate is simple Monod kinetic growth with a constant death rate. In (1.2) there is only a consumption term from the bacteria consuming the carbon substrait. 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 1.1.

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 1.1: List of parameters and their dimensions

1.3 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 substrait, M_∞ and C_∞. From using the following parameter changes, the system can be made unitless.

$$\chi = \frac{x}{L} \implies Ld\chi = dx \tag{1.6}$$

$$\tau = ut \implies \frac{1}{u}d\tau = dt \tag{1.7}$$

$$\mathcal{M} = \frac{M}{M_\infty} \tag{1.8}$$

$$\mathcal{C} = \frac{C}{C_\infty} \tag{1.9}$$

$$\delta = \frac{1}{uL^2}d \tag{1.10}$$

$$\kappa = \frac{k}{C_\infty} \tag{1.11}$$

$$\nu = \frac{n}{uC_\infty} \tag{1.12}$$

$$\gamma = \frac{M_\infty}{C_\infty}y \tag{1.13}$$

Using these, (1.1) and (1.2) can be simplified and nondimensionalized into,

$$\mathcal{M}_\tau = \nabla_\chi \left(D(\mathcal{M})\nabla_\chi \mathcal{M}\right) + F(\mathcal{C})\mathcal{M} \tag{1.14}$$

$$\mathcal{C}_\tau = -G(\mathcal{C})\mathcal{M}, \tag{1.15}$$

where,

$$\begin{aligned} D(\mathcal{M}) &= \delta \frac{\mathcal{M}^\alpha}{(1-\mathcal{M})^\beta} \\ F(\mathcal{C}) &= \frac{\mathcal{C}}{\kappa + \mathcal{C}} - \nu \\ G(\mathcal{C}) &= \gamma \frac{\mathcal{C}}{\kappa + \mathcal{C}}. \end{aligned} \tag{1.16}$$

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

1.4 Parameters

Each of the dimensionless parameters in (1.16) 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 substrait 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 substrait. Lastly, γ is the yield ratio. It signifies the ratio of substrait consumed to biomass growth. Here, a larger γ value results in more substrait being consumed to produce the same amount of biomass.

With (1.14) being reduced to four parameters the numerical analysis become more simiplified while still retaining the same significance in results.

Chapter 2

Numerical Methods

2.1 Discretization

In order to find the solution for (1.14) spatial and temporal discretizations must be made. First the equations are discretzied in time,

$$\frac{M^{k+1} - M^k}{\Delta t} = \nabla_x (D(M^{k+1}) \nabla_x M^{k+1}) + F(C^{k+1}) M^{k+1}, \tag{2.1}$$

$$\frac{C^{k+1} - C^k}{\Delta t} = \frac{h}{2}(G(C^{k+1}) M^{k+1} + G(C^k) M^k). \tag{2.2}$$

Here, (2.1) follows the ideas of the Backwards Euler Method; (2.2) follows Trapezoidal Rule. The index variable k has also been introduced in (2.1 - 2.2) 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 (2.1) 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. This means that the distance between grid points are the same in both x_1 and x_2 dimensions; we have $\Delta x_1 = \Delta x_2$. The solution of (2.1) will be approximated at each grid point using a five-point stencil. To index the grid point, i and j are used such that $M_{i,j}^{k} \approx M(t^k, x_{1_i}, x_{2_j})$. Because of the five-point stencil, the boundary gridpoints will depend on ghost grid points. This means that the equation to solve interior grid points will differ slightly from boundary points. The resulting equation for interior points, where $i \in (1, 2, \dots, n-1)$ and $j \in (1, 2, \dots, m-1)$, is

$$\frac{M_{i,j}^{k+1}-M_{i,j}^{k}}{\Delta t} = \frac{1}{\Delta x^2}\sum_{(s,r)\in\mathbb{A}}\left(\frac{D(M_{i+s,j+r}^{k+1})+D(M_{i,j}^{k+1})}{2}\cdot(M_{i+s,j+r}^{k+1}-M_{i,j}^{k+1})\right)+F(C_{i,j}^{k+1})M_{i,j}^{k+1} \tag{2.3}$$

where $\mathbb{A} = \{(0,\pm 1), (\pm 1, 0)\}$. The resulting equation for boundary points, when $(i,j) \in \{0, n\} \times \{0, 1, \dots, m\}$ or $(i,j) \in \{0, 1, \dots, n\} \times \{0, m\}$, is

$$\begin{aligned}\frac{M_{i,j}^{k+1}-M_{i,j}^{k}}{\Delta t} =& \frac{1}{\Delta x^2}\sum_{(s,r)\in\mathbb{A}}\left(\frac{D(M_{i+s,j+r}^{k+1})+D(M_{i,j}^{k+1})}{2}\cdot(M_{i+s,j+r}^{k+1}-M_{i,j}^{k+1})\right)\\ &+\sum_{(s,r)\in\mathbb{B}}\left(\frac{D(M_{i+s,j+r}^{k+1})+D(M_{i,j}^{k+1})}{2}\cdot(M_{i-s,j-r}^{k+1}-M_{i+s,j+r}^{k+1})\right)+F(C_{i,j}^{k+1})M_{i,j}^{k+1}\end{aligned} \tag{2.4}$$

where $\mathbb{B}$ depends on the boundary position; $\mathbb{B}_1 = \{(0,-1)\}$, $\mathbb{B}_2 = \{(-1,0)\}$, $\mathbb{B}_3 = \{(1,0)\}$, $\mathbb{B}_4 = \{(0,1)\}$. For the corner points, where two different boundaries connect, the result is to use a cross-product between $\mathbb{B}$'s, for example $\mathbb{B} = \mathbb{B}_1 \times \mathbb{B}_2$ For both (2.3 - 2.4) the arithmetic mean of the diffusion function, D, is taken because of its steep gradiant at the interface.

2.2 Solving Method

Now there exist equations for which C and M can be solved, (2.2) and (2.3 - 2.4) respectivly. Using C^k and $M_{i,j}^{k}$ as approximations of the solutions for (1.14) will allow the system to be solved by computing C^{k+1} and $M_{i,j}^{k+1}$. However, there are complications with trying to get an explicit formula for $M_{i,j}^{k+1}$ from (2.3 - 2.4) because of the dependency in $D(M)$ and $F(C)$. 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:

$$\frac{M_{i,j}^{(p+1)} - M_{i,j}^{k}}{\Delta t} = \frac{1}{\Delta x^2} \sum_{(s,r)\in\mathbb{A}} \left(\frac{D(M^{p}_{i+s,j+r}) + D(M^{p}_{i,j})}{2} \cdot (M^{(p+1)}_{i+s,j+r} - M^{(p+1)}_{i,j}) \right) + F(C^{(p)}_{i,j}) M^{(p+1)}_{i,j} \tag{2.5}$$

$$\frac{C^{(p+1)} - C^{k}}{\Delta t} = \frac{-1}{2} (G(C^{(p+1)}) M^{(p+1)} + G(C^{k}) M^{k}) \tag{2.6}$$

where $p \in (0, 1, \ldots, P)$. It is important to show explicitly the the purpose of the fixed point iteration is to link two distinct times with P solutions in between them, such that:

$$\begin{aligned} M^{(p=0)} &= M^{k}, \quad M^{(p=P)} = M^{k+1}, \\ C^{(p=0)} &= C^{k}, \quad C^{(p=P)} = C^{k+1}. \end{aligned} \tag{2.7}$$

In this fixed point formate, given by (2.5 - 2.6), the equations can be rearrange and solved by conventional methods.

For (2.5), a linear system of equations can be created following Saad (2003). For each grid point (i, j) a linear system exists, defined as:

$$\begin{aligned} \frac{M^{k}_{i,j}}{\Delta t} = &\sum_{(i,j)\in\mathbb{A}} \left(\frac{D(M^{p+1}_{i+s,j+r}) + D(M^{p+1}_{i,j})}{2\Delta x^2} \cdot M^{p+1}_{i+s,j+r} \right) \\ &+ \left(\sum_{(i,j)\in\mathbb{A}} \left(\frac{D(M^{p+1}_{i+s,j+r}) + D(M^{p+1}_{i,j})}{2\Delta x^2} \right) - F(C^{p}_{i,j}) + \frac{0}{\Delta t} \right) M^{p+1}_{i,j}. \end{aligned} \tag{2.8}$$

From (2.8), a five-diagonal matrix can be created defined as,

$$A = \begin{pmatrix} a_{i,j} & a_{i+1,j} & & a_{i,j+1} & & & & & \\ a_{i-1,j} & \ddots & \ddots & & \ddots & & & & \\ & \ddots & \ddots & \ddots & & \ddots & & & \\ a_{i,j-1} & & a_{i-1,j} & a_{i,j} & a_{i+1,j} & & a_{i,j+1} & & \\ & \ddots & & \ddots & \ddots & \ddots & & \ddots & \\ & & a_{i,j-1} & & a_{i-1,j} & a_{i,j} & a_{i+1,j} & & a_{i,j+1} \\ & & & \ddots & & \ddots & \ddots & \ddots & \\ & & & & \ddots & & \ddots & \ddots & a_{i+1,j} \\ & & & & & a_{i,j-1} & & a_{i-1,j} & a_{i,j} \end{pmatrix} \tag{2.9}$$

where each $a_{i,j}$ is the coefficient based on (2.8).

Solving (2.9) can be done by use of the Conjugate Gradient method provided that certain conditions are satisfied.

Proposition 2.2.1. *The matrix A, defined in (2.9) is positive definite and symmetric when $\frac{1}{F(C^{(p)}_{i,j})} < \Delta t$.*

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 (2.8) it can be said that,

$$\sum_{(i,j)\in\mathbb{A}} \left(\frac{D(M^p_{i+\frac{s}{2},j+\frac{r}{2}})}{\Delta x^2} \right) < \left(\sum_{(i,j)\in\mathbb{A}} \left(\frac{D(M^p_{i+\frac{s}{2},j+\frac{r}{2}})}{\Delta x^2} \right) - F(C^p_{i,j}) + \frac{1}{\Delta t} \right). \tag{2.10}$$

This simplifies to,

$$F(C^p_{i,j}) < \frac{1}{\Delta t} \tag{2.11}$$

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 subsequentally 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 (2.6), the equation can be rearranged into a quadratic form, substituting in $G(C)$ from (1.16)

$$\left(C^{p+1}\right)^2 + \left(\kappa - C^k + \frac{h}{2}\gamma M^{p+1} + \frac{h}{2}\frac{\gamma C^k M^k}{\kappa + C^k}\right) C^{p+1} + \left(-\kappa C^k + \frac{h}{2}\frac{\gamma \kappa C^k M^k}{\kappa + C^k}\right) = 0. \tag{2.12}$$

Using the quadratic equation results in,

$$C^{p+1} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{2.13}$$

for which,

$$\begin{aligned} a &= 1 \\ b &= \kappa - C^k + \frac{h}{2}\gamma M^{p+1} + \frac{h}{2}\frac{\gamma C^k M^k}{\kappa + C^k} \\ c &= -\kappa C^k + \frac{h}{2}\frac{\gamma\kappa C^k M^k}{\kappa + C^k} \end{aligned} \tag{2.14}$$

To determine which branch of (2.13) to use, a physical situation is used. Specifically the case where there exist no biomass, $M = 0$. The expected outcome is that no substrate is consumed and thus the substrait concentration will remain constant as a function of t. When the equations in (2.14) are evaluated at $M = 0$, the result it,

$$a = 1, \quad b = \kappa - C^k, \quad c = -\kappa C^k, \tag{2.15}$$

which can be used to evaluate (2.13) as,

$$\begin{aligned} C^{p+1} &= \frac{-(\kappa - C^k) \pm \sqrt{(\kappa - C^k)^2 - 4(-\kappa C^k)}}{2} \\ &= \frac{1}{2}\left(C^k - \kappa \pm \sqrt{\kappa^2 + 2\kappa C^k + \left(C^k\right)^2}\right) \\ &= \frac{1}{2}\left(C^k - \kappa \pm (\kappa + C^k)\right). \end{aligned} \tag{2.16}$$

Now, if the positive branch is used the above equation evaluates to $C^{p+1} = C^k$. This means that between any two distinct times, the substrait concentration will remain constants, which was expected. To further this confirmation, the negative branch results in $C^{p+1} = -\kappa$, a non-postive substrate concentration, which is not physically relavent.

$$C^{p+1} = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \tag{2.17}$$

where a, b, and c are defined in (2.14).

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 esstablished. Algorithm 1 shows the organizations of solving (2.6

- 2.5). Note that Algorithm 1 actually describes both a fully- and semi- implicite method for solving

```
Data: $M^p$ and $C^p$ are temporary solutions defined such that
    $M^p \to M^{k+1}$ and $C^p \to C^{k+1}$ as $p \to P$.
    $\epsilon_{sol}$ is a tolerance set for a desired accuracy.
begin
    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
end
```

Algorithm 1: Algorithm for the fully-implicit solving of (1.14)

(1.14). If $P = 1$ then only a single iteration of the algorithm is applied. 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:

$$\begin{array}{rccc} \pi: & \{0,\dots,n\} \times \{0,\dots,m\} & \to & \{1,\dots,nm\} \\ & (i,j) & \to & \pi(i,j) \end{array} \tag{2.18}$$

This mapping allows the system to be easily stored in diagonal format, since (2.9) has five distinct diagonals.

2.3 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
```

2.4 Numerical Results

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 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 approche a single solution as they become more precise. This convergence test will also show the thresholds for an accurate simulation result, to help reduce the computiation times. With a well-established method, the comparison between semi- and fully-implicit methods can be done.

2.4.1 Basic Simulations

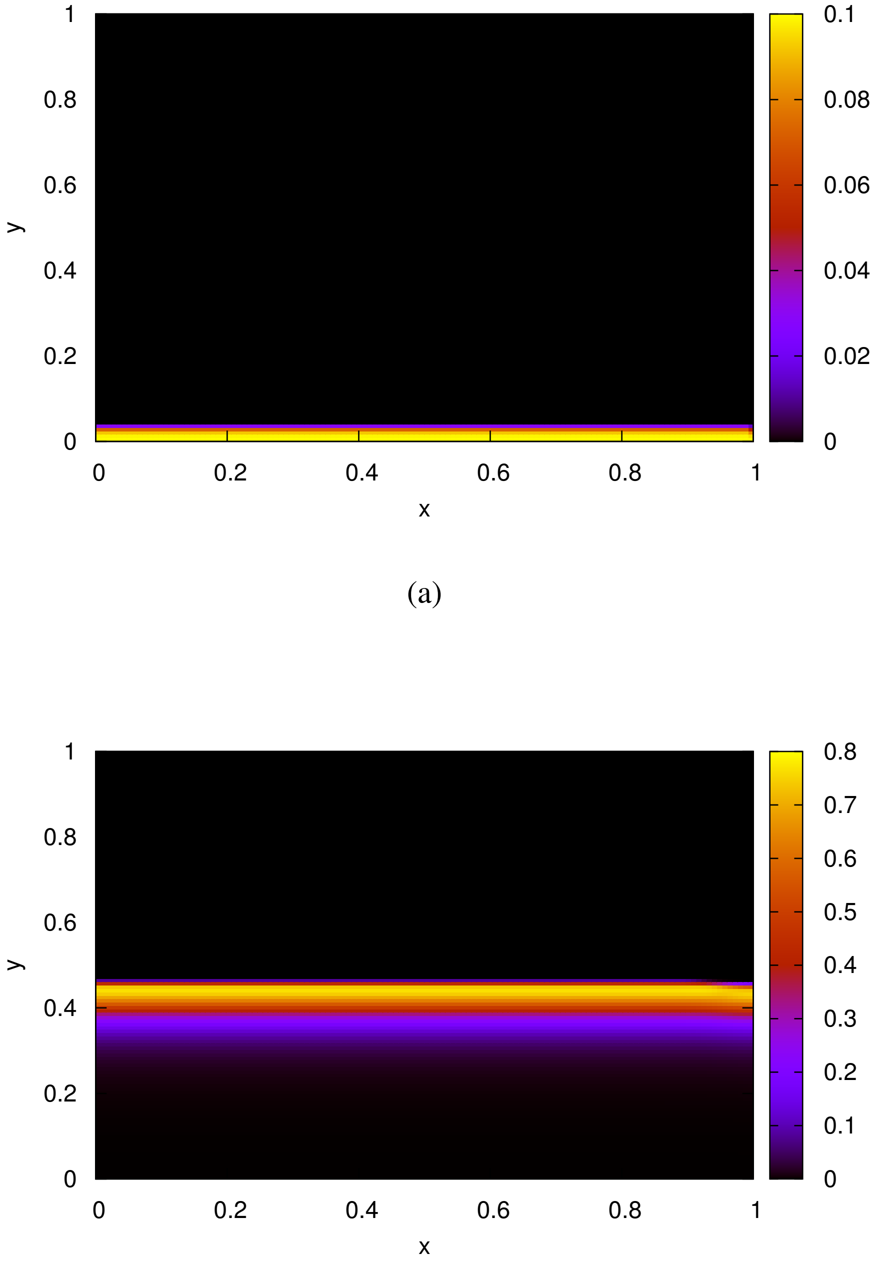
Using Algorithm 1, simple scenarios can be tested as a first verification on the method.

A simple test would test if the diffusion reaction model works for the propagation of the biomass. Having, all of the biomass on one boundary of Ω would show the spatial diffusion across the region. With the initial conditions,

$$
\begin{aligned}
M &= \begin{cases} -\left(\frac{h}{d^4}\right)x^4 + h & \text{if } y \leq d \\ 0 & \text{otherwise} \end{cases} \\
C &= 1
\end{aligned}
\tag{2.19}
$$

where $h = 0.1$ and $d = \frac{5}{128}$. Here, h and d represent the height and depth of the innoculation site. The solution in Figure 2.1 show that, with initial conditions given in (2.19), the solution propagate consistently with time.

The total amount of biomass has no known exact value, given the choice of growth rate function. However, the amount of total biomass can be predicted to grow exponentially if the growth rate



(a)

(b)

Figure 2.1: Solutions for M with initial conditions given in (2.19) at (a) $t = 0$ and (b) $t = 40$.

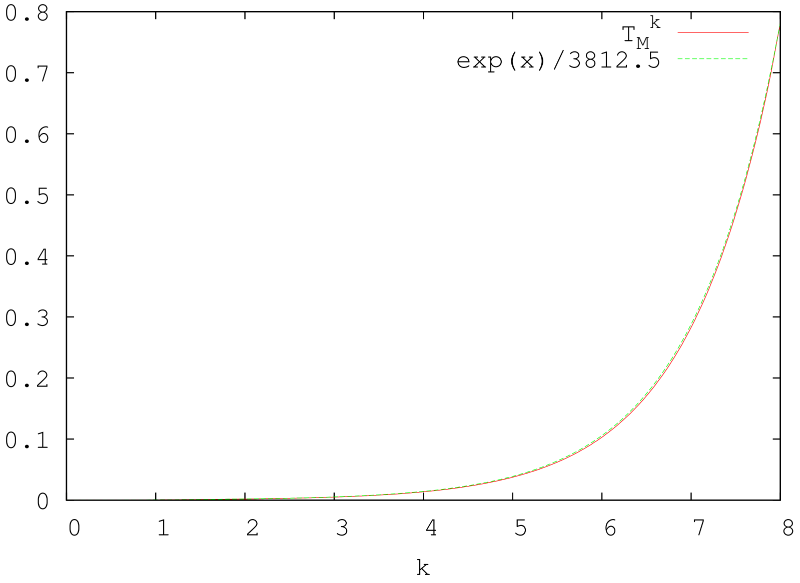


Figure 2.2: Plot of total biomass, T_M^k, for $k = 0, 1, ..., 20$.

function, $F(C)$ from (1.16), was replaced with a constant. This can be checked by tracking the total biomass, called $T_M(t)$, with the changed growth rate function:

$$F(C) = a \tag{2.20}$$

Tracking $T_M(t)$ can be done by,

$$T_M(t) = \int_\Omega M(t)dx. \tag{2.21}$$

Numerically, this is computed by grid-wise summation,

$$T_M(t^k) \approx T_M^k = \frac{\sum_i^n \sum_j^m M_{i,j}^k}{nm} \tag{2.22}$$

Figure 2.2 shows that the total biomass is as expected, since it matchs the curve of $\frac{1}{3812.5} * e^k$, an exponential function.

2.4.2 Convergence 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.

2.4.2.1 Error Computations

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

$$\epsilon_{sol} = \frac{||u_1 - u_2||}{||u_2||} \tag{2.23}$$

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 that 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 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 normed used during the computations. Here three norms will be used:

$$\ell_1 : ||u||_1 = \sum_{\pi(i,j)}^{nm} |u_{i,j}| \tag{2.24}$$

$$\ell_2 : ||u||_2 = \sqrt{\sum_{\pi(i,j)}^{nm} (u_{i,j})^2} \tag{2.25}$$

$$\ell_\infty : ||u||_\infty = \max_{\substack{i=1,\dots,n \\ j=1,\dots,m}} |u_{i,j}| \tag{2.26}$$

These different norms will all be used to create a broader understanding of the error. This creates three

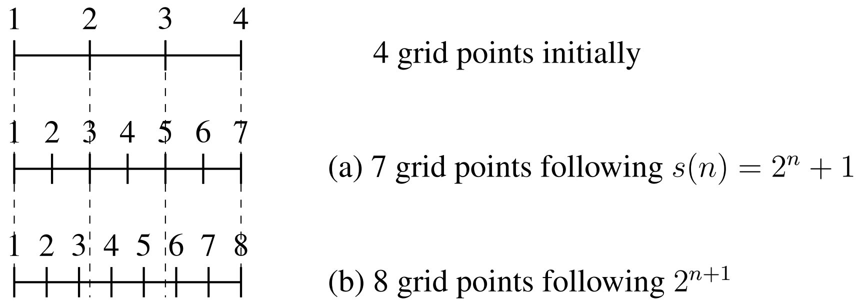

1 2 3 4
4 grid points initially
1 2 3 4 5 6 7
(a) 7 grid points following $s(n) = 2^n + 1$
1 2 3 4 5 6 7 8
(b) 8 grid points following 2^{n+1}

Figure 2.3: Visualization in 1D explaining the use of $s(n) = 2^n + 1$ instead of 2^{n+1} to control the grid size selection. When the number of grid points are (b) doubled they do not lineup with the grid points of the previous discritization. With (a) this problem does not exist.

distinct values for ϵ_{sol}, named ϵ_{ℓ_1}, ϵ_{ℓ_2}, and ϵ_{ℓ_∞}; each named for the norm used during the computation.

2.4.2.2 Grid 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 (2.19) due to the simplicity. The convergence will be tracked with all three forms of ϵ_{sol} Since there exist a difference in the number of grid points between the different solutions, u_1 and u_2, only the grid points in the more coarse refinement will be used. This places a limitation on the selection of grid-sizes and so they will follow the series $s(n) = 2^{n-1} + 1$ for $n \in \mathbb{Z}$. So that the grid points can line up, without any use of linear interpolation, grid refine must be using $s(n)$ instead of the typical grid doubling of 2^n. This is illustrated in Figure 2.3.

The results from Figure 2.4 show that the solutions converge as the grid size becomes refined.

2.4.3 Results

Here the main comparison that analysis the effects of using Algorithm 1 with different P values. Recall that the main observation is for $P = 1$, which corelates to the semi-implicit method. Along with accuracy, the simulation time will be 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 P increases. Therefore, this needs

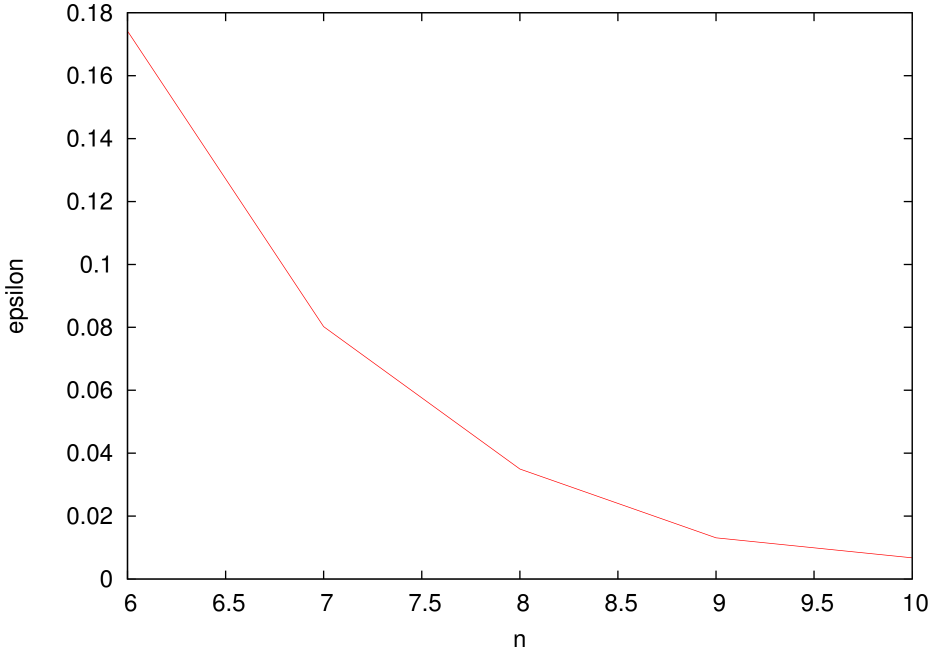


Figure 2.4: Plot showing the convergence of solutions based on changed in Δx. The computions are of ϵ_{ℓ_1}, ϵ_{ℓ_2}, and ϵ_{ell_∞} with grid-size following $s(n) = 2^{n-1} + 1$.

284 to be weighted against the cost of computational intensity and the increase of the simulation time.

285 The results of the method comparison can be seen in Table 2.1.

P	Simulation Time	ϵ_{ℓ_1}	ϵ_{ℓ_2}	ϵ_{ℓ_∞}
1	60.05	0.0018	-	-
2	97.72	0.0014	-	-
3	117.46	0.0013	-	-
4	126.44	0.0011	-	-

Table 2.1: Results from running simulation with different P values. Note, $P = 1$ corresponds to the semi-implicit method.

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.