

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

© E.M. Jalbert, January, 2015

¹⁵ **Contents**

¹⁶ 1	Model Definition	1
¹⁷	1.1 History	1
¹⁸	1.2 Model Description	2
¹⁹	1.3 Nondimensionalization	3
²⁰	1.4 Parameters	5
²¹ 2	Numerical Methods	6
²²	2.1 Discretization	6
²³	2.2 Solving Method	7
²⁴	2.3 Computational Setup	12
²⁵	2.4 Numerical Results	13
²⁶	Complete Bibliography	20

²⁷ **List of Figures**

28	2.1	Solutions for (ac) M and (bd) C with 1D initial conditions defined in (2.19) at (ab) $t = 0$ and (cd) $t = 40$. Computed with a 1025×1025 grid and a timestep of $\Delta t = 10^{-3}$.	14
29	2.2	Solutions for M with spherical initial conditions defined by (2.20) at (a) $t = 0$ and (b) $t = 40$. Computed with a 1025×1025 grid and a timestep of $\Delta t = 10^{-3}$.	15
30	2.3	Plot of the relative error between the computed total biomass, $T_M(t)$, and the the- oretical total biomass, $y_0 e^x$. 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.	16
31	2.4	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.	18
32	2.5	Plot showing the convergence of solutions based on changed in Δx . The computations are of ϵ_{ℓ_1} , ϵ_{ℓ_2} , and ϵ_{ell_∞} with grid-size following $s(n) = 2^{n-1} + 1$.	19

⁴² List of Tables

⁴³	1.1 List of parameters and their dimensions	3
⁴⁴	2.1 Results from running simulation with different P values. Note, $P = 1$ corresponds	
⁴⁵	to the semi-implicit method.	19

⁴⁶ **Chapter 1**

⁴⁷ **Model Definition**

⁴⁸ **1.1 History**

⁴⁹ The tradition biofilm model has been continually developed over many iterations since 1980. Rittmann
⁵⁰ and McCarty (1980) formulated the steady-state biofilm model, developed using the concept that
⁵¹ biofilm growth would be the result of a steady flux from substrate. Since then the model have evolved
⁵² to include modelling three-dimensional growth of multispecies anaerobic biofilms (Noguera et al.
⁵³ (1999)) and spatially heterogeneous biofilm structures (Eberl et al. (2001)).

⁵⁴ The modelling of *Clostridium Thermocellum* is unique because this cellulolytic anaerobic bacteria
⁵⁵ does not generate an extracellular polymeric substance. This uncharacteristic behaviour means that
⁵⁶ the mathematical model based on the work of Eberl and Demaret (2007) cannot be used as is. They
⁵⁷ modelled the biomass density and nutrient concentrations as a two-PDE-coupled system. Recently,
⁵⁸ Wang et al. (2011), used a cellular automata based model for simulating the growth of *Clostridium*
⁵⁹ *Thermocellum*. From this, better results were thought to derive from a continuous differential equa-
⁶⁰ tion based model. Here the spatial diffusion of the substrait concentration is removed to mimic the
⁶¹ carbon substrait that is consumed by *Clostridium Thermocellum*. This results in a PDE-ODE-coupled
⁶² system. This is based on the work done by Dumitrache (2014), where this same coupling was used
⁶³ and formulated.

64 1.2 Model Description

65 The model used for simulations is based on the deterministic model developed in Eberl et al. (2001),
 66 which was designed to simulate the development of spatially heterogeneous biofilm structures. They
 67 modelled the biomass density and nutrient concentration as a two-PDE-coupled system. Here the spa-
 68 tial diffusion of the nutrient concentration is removed to mimic the carbon substrate that *C.Thermocellum*
 69 consumes in growth. This makes a PDE-ODE-coupled system purposed as,

$$M = \nabla_x (d(M) \nabla_x M) + f(C)M \quad (1.1)$$

$$C = -g(C)M \quad (1.2)$$

70 where

$$71 d(M) = d \frac{M^\alpha}{(1 - M)^\beta} \quad (1.3)$$

$$72 f(C) = u \frac{C}{k + C} - n \quad (1.4)$$

$$74 75 g(C) = y \frac{C}{k + C} \quad (1.5)$$

76 Here we have a pair of equations, (1.1) and (1.2), that represent the biomass density and substrate
 77 concentration respectively. The spatial diffusion of the biofilm is modelled with density-dependent
 78 diffusion, represented by (1.3), and the growth rate of biomass is given by (1.4). The growth rate is
 79 simple Monod kinetic growth with a constant death rate. In (1.2) there is only a consumption term
 80 from the bacteria consuming the carbon substrate. This term is based on the growth of the biomass,
 81 differing only by a scalar multiplier.

82 The dimensions of the parameters and variables are in Table 1.1.

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

Table 1.1: List of parameters and their dimensions

83 1.3 Nondimensionalization

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

$$\chi = \frac{x}{L} \implies L d\chi = dx \quad (1.6)$$

$$\tau = ut \implies \frac{1}{u} d\tau = dt \quad (1.7)$$

$$\mathcal{M} = \frac{M}{M_\infty} \quad (1.8)$$

$$\mathcal{C} = \frac{C}{C_\infty} \quad (1.9)$$

$$\delta = \frac{1}{uL^2} d \quad (1.10)$$

$$\kappa = \frac{k}{C_\infty} \quad (1.11)$$

$$\nu = \frac{n}{uC_\infty} \quad (1.12)$$

$$\gamma = \frac{M_\infty}{C_\infty} y \quad (1.13)$$

⁸⁹ Using these, (1.1) and (1.2) can be simplified and nondimensionalized into,

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

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

⁹⁰ where,

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

⁹² with only $\delta, \kappa, \nu, \gamma$ as model parameters.

93 1.4 Parameters

94 Each of the dimensionless parameters in (1.16) have a biological representation based on the transfor-
95 mations done. The parameter δ is the dimensionaless constant for diffusion. It affects the change in
96 biomass from adjacent biomass sources, a greater δ results in a greater change. The parameter κ is the
97 half-saturation point, it is exactly the value for which substrait concentration results in 0.5-optimum
98 growth rate. Parameter ν is the death rate of the biomass. Specifically, it is the ratio of biomass
99 growth to death, representing the fraction of biomass density that perishes from natural causes or a
100 lack of substrait. Lastly, γ is the yield ratio. It signifies the ratio of substrait consumed to biomass
101 growth. Here, a larger γ value results in more substrait being consumed to produce the same amount
102 of biomass.

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

105 **Chapter 2**

106 **Numerical Methods**

107 **2.1 Discretization**

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

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

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

113 Here, (2.1) follows the ideas of the Backwards Euler Method; (2.2) follows Trapezoidal Rule. The
114 index variable k has also been introduced in (2.1 - 2.2) such that $M^k(x) \approx M(t^k, x)$, allowing an
115 approximation at a certain time, t^k , to be used; this reduces the dimensionality of the problem.

116 For this system, the region of consideration will be a rectangular region, Ω . This region has Neumann
117 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
118 since, according to the biology of our system, the substrate does not diffuse across the region. The
119 spatial discretization will be through the Finite Difference Method as described in Saad (2003). Here,
120 a uniform $n \times m$ grid is used to discretize Ω . This means that the distance between grid points
121 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

122 approximated at each grid point using a five-point stencil. To index the grid point, i and j are used
 123 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
 124 depend on ghost grid points. This means that the equation to solve interior grid points will differ
 125 slightly from boundary points. The resulting equation for interior points, where $i \in (1, 2, \dots, n - 1)$
 126 and $j \in (1, 2, \dots, m - 1)$, is

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

127 where $\mathbb{A} = \{(0, \pm 1), (\pm 1, 0)\}$. The resulting equation for boundary points, when $(i, j) \in \{0, n\} \times$
 128 $\{0, 1, \dots, m\}$ or $(i, j) \in \{0, 1, \dots, n\} \times \{0, m\}$, is

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

130 where \mathbb{B} depends on the boundary position; $\mathbb{B}_1 = \{(0, -1)\}$, $\mathbb{B}_2 = \{(-1, 0)\}$, $\mathbb{B}_3 = \{(1, 0)\}$,
 131 $\mathbb{B}_4 = \{(0, 1)\}$. For the corner points, where two different boundaries connect, the result is to use a
 132 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
 133 diffusion function, D , is taken because of its steep gradient at the interface.

135 2.2 Solving Method

136 Now there exist equations for which C and M can be solved, (2.2) and (2.3 - 2.4) respectively. Using
 137 C^k and $M_{i,j}^k$ as approximations of the solutions for (1.14) will allow the system to be solved by
 138 computing C^{k+1} and $M_{i,j}^{k+1}$. However, there are complications with trying to get an explicit formula
 139 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
 140 point iteration is introduced. In a single time step, the solutions for M and C can be solved using the

¹⁴¹ previous time step solution in the follow manner:

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

¹⁴² (2.5)

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

¹⁴⁴ (2.6)

¹⁴⁵ where $p \in (0, 1, \dots, P)$. It is important to show explicitly the the purpose of the fixed point iteration

¹⁴⁶ is to link two distinct times with P solutions in between them, such that:

$$\begin{aligned} M^{(p=0)} &= M^k, & M^{(p=P)} &= M^{k+1}, \\ \text{147 } C^{(p=0)} &= C^k, & C^{(p=P)} &= C^{k+1}. \end{aligned}$$

(2.7)

¹⁴⁸ In this fixed point formate, given by (2.5 - 2.6), the equations can be rearrange and solved by conventional methods.

¹⁵⁰ For (2.5), a linear system of equations can be created following Saad (2003). For each grid point (i, j)
¹⁵¹ a linear system exists, defined as:

$$\begin{aligned} \frac{M_{i,j}^k}{\Delta t} &= \sum_{(i,j) \in \mathbb{A}} \left(\frac{D(M_{i+s,j+r}^{p+1}) + D(M_{i,j}^{p+1})}{2\Delta x^2} \cdot M_{i+s,j+r}^{p+1} \right) \\ \text{152 } &+ \left(\sum_{(i,j) \in \mathbb{A}} \left(\frac{D(M_{i+s,j+r}^{p+1}) + D(M_{i,j}^{p+1})}{2\Delta x^2} \right) - F(C_{i,j}^p) + \frac{0}{\Delta t} \right) M_{i,j}^{p+1}. \end{aligned}$$

(2.8)

¹⁵³ From (2.8), a five-diagonal matrix can be created defined as,

$$\text{154 } A = \begin{pmatrix} a_{i,j} & a_{i+1,j} & & a_{i,j+1} & & \\ a_{i-1,j} & \ddots & \ddots & & \ddots & \\ & \ddots & \ddots & \ddots & & \ddots \\ a_{i,j-1} & & a_{i-1,j} & a_{i,j} & a_{i+1,j} & a_{i,j+1} \\ & \ddots & & \ddots & \ddots & \ddots & \ddots \\ & & a_{i,j-1} & a_{i-1,j} & a_{i,j} & a_{i+1,j} & a_{i,j+1} \\ & & & \ddots & \ddots & \ddots & \ddots \\ & & & & \ddots & \ddots & \ddots & a_{i+1,j} \\ & & & & & a_{i,j-1} & a_{i-1,j} & a_{i,j} \end{pmatrix} \quad (2.9)$$

¹⁵⁵ where each $a_{i,j}$ is the coefficient based on (2.8).

¹⁵⁶ Solving (2.9) can be done by use of the Conjugate Gradient method provided that certain conditions
¹⁵⁷ are satisfied.

¹⁵⁸ **Proposition 2.2.1.** *The matrix A , defined in (2.9) is positive definite and symmetric when $\frac{1}{F(C_{i,j}^{(p)})} <$*
¹⁵⁹ Δt .

¹⁶⁰ *Proof.* Matrix A is positive definite if all the eigenvalues are positive. Using the Circle theorem
¹⁶¹ described by Geršgorin (1931), the eigenvalues can be shown to be positive if, independently on all
¹⁶² rows, the sum of the off-diagonals values is less than the diagonal value. This can be verified. From
¹⁶³ (2.8) it can be said that,

$$\text{164 } \sum_{(i,j) \in \mathbb{A}} \left(\frac{D(M_{i+\frac{s}{2}, j+\frac{r}{2}}^p)}{\Delta x^2} \right) < \left(\sum_{(i,j) \in \mathbb{A}} \left(\frac{D(M_{i+\frac{s}{2}, j+\frac{r}{2}}^p)}{\Delta x^2} \right) - F(C_{i,j}^p) + \frac{1}{\Delta t} \right). \quad (2.10)$$

¹⁶⁵ This simplifies to,

$$\text{166 } F(C_{i,j}^p) < \frac{1}{\Delta t} \quad (2.11)$$

167 The symmetry of A can be trivially shown if one considers the formation of the diagonals. On a
 168 single row, each element corresponds to the adjacent grid points of grid i, j . As the grid ordering
 169 counts along, the elements that are equidistance from the diagonal are actually reference to the same
 170 grid point. Therefore we have symmetry. \square

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

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

183 For solving (2.6), the equation can be rearranged into a quadratic form, substituting in $G(C)$ from
 184 (1.16)

$$185 \quad (C^{p+1})^2 + \left(\kappa - C^k + \frac{h}{2} \gamma M^{p+1} + \frac{h}{2} \frac{\gamma C^k M^k}{\kappa + C^k} \right) C^{p+1} + \left(-\kappa C^k + \frac{h}{2} \frac{\gamma \kappa C^k M^k}{\kappa + C^k} \right) = 0. \quad (2.12)$$

186 Using the quadratic equation results in,

$$187 \quad C^{p+1} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (2.13)$$

188 for which,

$$a = 1$$

189

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

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

194

$$a = 1, \quad b = \kappa - C^k, \quad c = -\kappa C^k, \quad (2.15)$$

195 which can be used to evaluate (2.13) as,

196

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

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

201

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

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

203 Now that computable solutions for M and C at a single time step have been found, an algorithm to
204 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- implicit method for solving

Data: M^p and C^p are temporary solutions defined such that

$M^p \rightarrow M^{k+1}$ and $C^p \rightarrow C^{k+1}$ as $p \rightarrow P$.

ϵ_{sol} is a tolerance set for a desired accuracy.

```

begin
  while convergence is not achieved do
    Solve  $A^{(p)} M^{(p+1)} = b^{(p)}$ ;
    Solve  $C^{(p+1)} = \frac{1}{2} (2b \pm \sqrt{b^2 - 4c})$ ;
    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)

205
206 (1.14). If $P = 1$ then only a single iteration of the algorithm is applied. This would result in a change
207 similar to how the Gauss-Seidal method changes the Jacobi method; the values used would no longer
208 be updated in a single timestep when $P = 1$.
209 To use the algorithm, the matrix system was converted into a 1D array by use of a bijective mapping
210 defined as:

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

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

214 2.3 Computational Setup

215 The implementation of Algorithm 1 was done with Fortran.

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

220 computations; the compiler arguments were

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

222 2.4 Numerical Results

223 With a defined method and computational setup a variety of simulations can be run to observe the
 224 accuracy and behaviour of the method. An examination a typical simulation will show if the expected
 225 behaviour is observed, validating the method as functioning. A convergence analysis for the method
 226 can be done to confirm that solutions from different grid sizes approach a single solution as they
 227 become more precise. This convergence test will also show the thresholds for an accurate simula-
 228 tion result, to help reduce the computation times. With a well-established method, the comparison
 229 between semi- and fully-implicit methods can be done.

230 2.4.1 Basic Simulations

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

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

$$236 M = \begin{cases} -\left(\frac{h}{d^4}\right)x^4 + h & , \text{if } y \leq d \\ 0 & , \text{otherwise} \end{cases} \quad (2.19)$$

$$C = 1$$

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

238 The solution shown in Figure 2.1 shows that the 1D characteristic of the biomass stays at a later time.

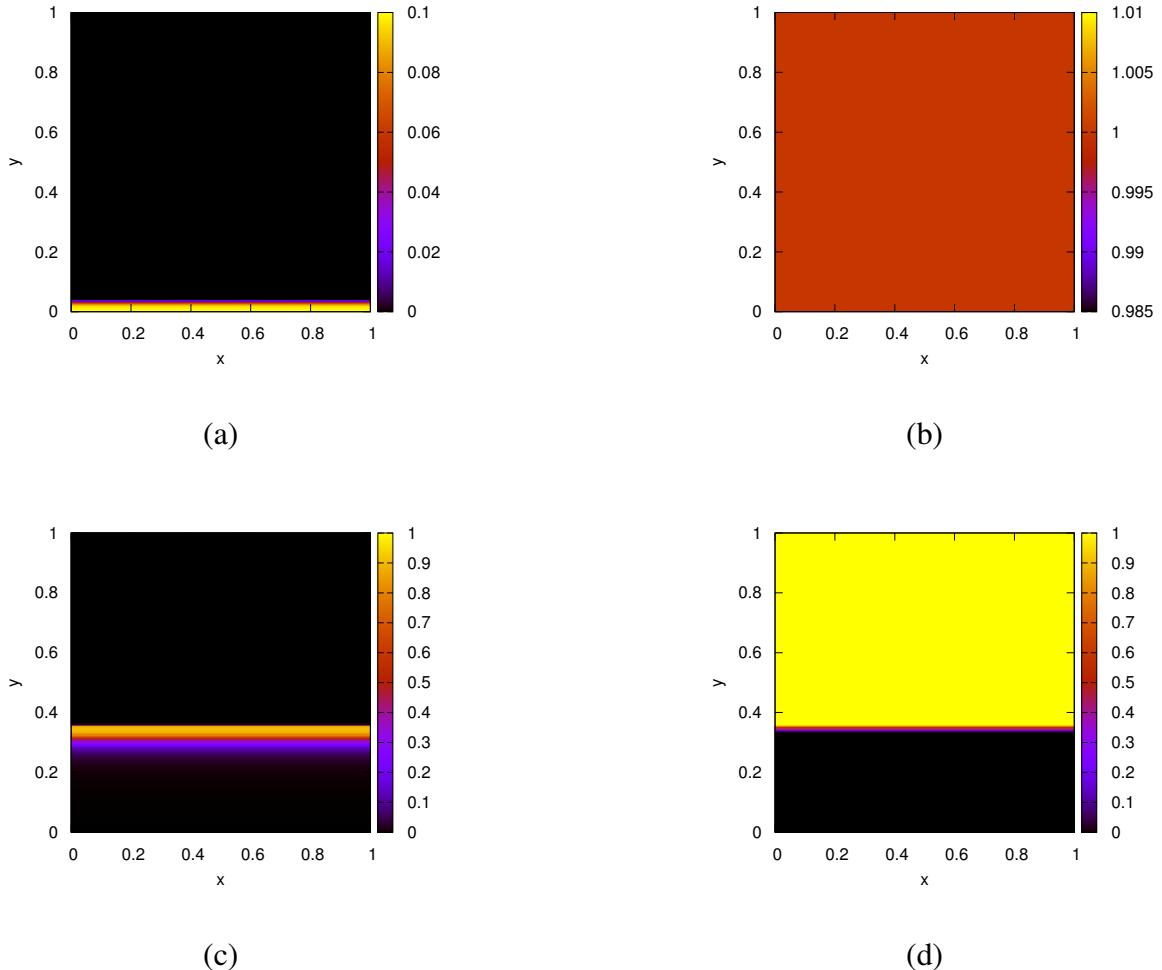


Figure 2.1: Solutions for (ac) M and (bd) C with 1D initial conditions defined in (2.19) at (ab) $t = 0$ and (cd) $t = 40$. Computed with a 1025×1025 grid and a timestep of $\Delta t = 10^{-3}$.

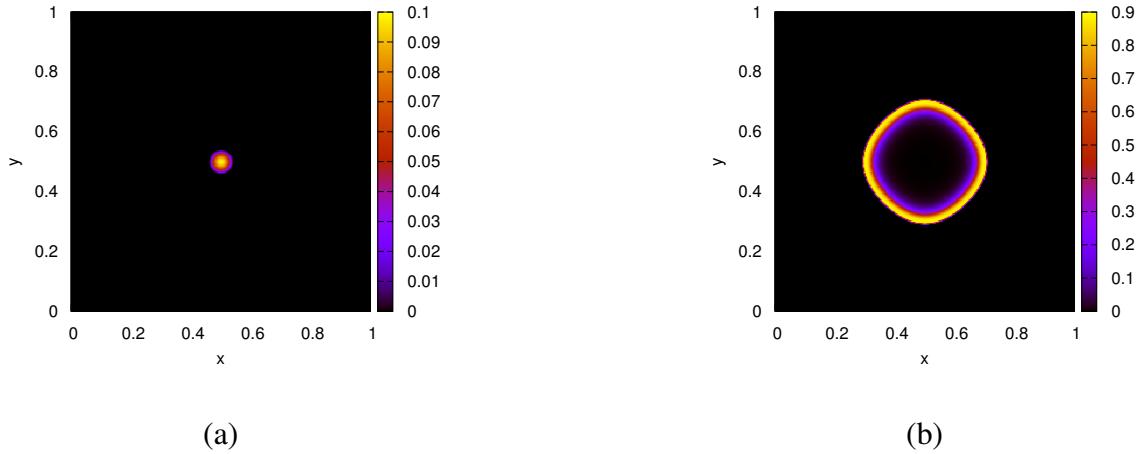


Figure 2.2: Solutions for M with spherical initial conditions defined by (2.20) at (a) $t = 0$ and (b) $t = 40$. Computed with a 1025×1025 grid and a timestep of $\Delta t = 10^{-3}$.

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

$$241 \quad M = \begin{cases} -\frac{h}{d^2} (x - 0.5)^2 + (y - 0.5)^2 + h & , \text{if } (x - 0.5)^2 + (y - 0.5)^2 < d^2 \\ 0 & , \text{otherwise} \end{cases}, \quad (2.20)$$

242 a test can be tried to see if the spherical nature of the solution is kept as time progresses. This can be
243 seen in Figure 2.2.

244 The given boundary conditions and spatial discretization could be a possible source or sink of biomass.
245 To ensure this is not the case, the total amount of biomass can be tracked. The total biomass cannot
246 be exactly determined with the given growth rate function. This means that there will not be anything
247 to measure the validity of the simulation solution against. However, if the growth rate were constant,
248 the total biomass would be expected to grow exponentially. This can be checked by tracking the total
249 biomass, now called $T_M(t)$, with the changed growth rate function:

$$250 \quad F(C) = 1 \quad (2.21)$$

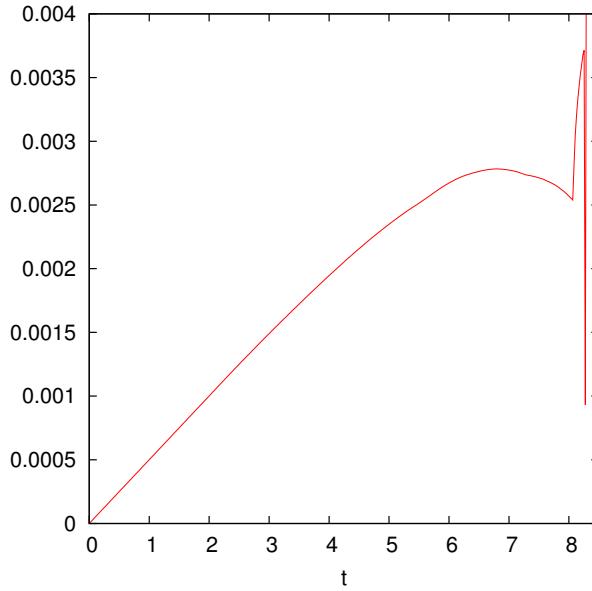


Figure 2.3: Plot of the relative error between the computed total biomass, $T_M(t)$, and the theoretical total biomass, $y_0 e^x$. The changes after $t = 8$ are from the biomass having completely filled the region Ω . This means that there is no physical space for the biomass to occupy and thus the growth slows down to a stop.

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

252
$$T_M(t) = \int_{\Omega} M(t) dx. \quad (2.22)$$

253 Numerically, this is computed by grid-wise summation,

254
$$T_M(t^k) \approx T_M^k = \frac{\sum_i^n \sum_j^m M_{i,j}^k}{nm} \quad (2.23)$$

255 Figure 2.3 shows that the total biomass is as expected, since it matches the curve of $\frac{1}{3812.5} * e^k$, an
256 exponential function. The simulation results in Figure 2.3 are from the same as Figure 2.2 but at later
257 times so that the boundary effects can be observed.

258 **2.4.2 Convergence Analysis**

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

263 **2.4.2.1 Error Computations**

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

266
$$\epsilon_{sol} = \frac{\|u_1 - u_2\|}{\|u_2\|} \quad (2.24)$$

267 where u_1 represents one simulation solution and u_2 represents the solution that is theoretically more
268 accurate. The theoretical accuracy of u_2 derives from the fact that most comparisons will be done
269 between solutions that vary in only Δx or between semi- and fully- implicit. These are understood to
270 have the relation that a smaller Δx , and that the fully-implicit method is to be more accurate. There
271 is an assumption that both u_1 and u_2 have the same number of grid points, so that the difference can
272 be taken grid-wise.

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

275
$$\ell_1 : \|u\|_1 = \sum_{\pi(i,j)}^{nm} |u_{i,j}| \quad (2.25)$$

276
$$\ell_2 : \|u\|_2 = \sqrt{\sum_{\pi(i,j)}^{nm} (u_{i,j})^2} \quad (2.26)$$

278
$$\ell_\infty : \|u\|_\infty = \max_{\substack{i=1,\dots,n \\ j=1,\dots,m}} |u_{i,j}| \quad (2.27)$$

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

282 **2.4.2.2 Grid Size Convergence**

283 To observe the validity of the method, a test on the convergence of solutions based on the spatial
284 discretization is done. This will involve using the same simulation described in (2.19) due to the
285 simplicity. The convergence will be tracked with all three forms of ϵ_{sol} . Since there exist a difference

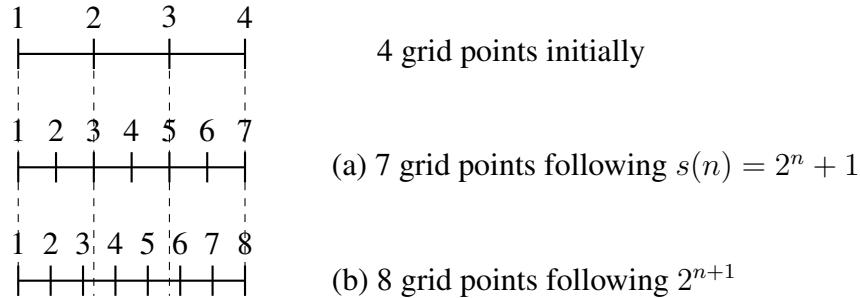


Figure 2.4: Visualization in 1D explaining the use of $s(n) = 2^n + 1$ instead of 2^{n+1} to control the grid size selection. When the number of grid points are (b) doubled they do not lineup with the grid points of the previous discretization. With (a) this problem does not exist.

286 in the number of grid points between the different solutions, u_1 and u_2 , only the grid points in the
 287 more coarse refinement will be used. This places a limitation on the selection of grid-sizes and so
 288 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
 289 any use of linear interpolation, grid refine must be using $s(n)$ instead of the typical grid doubling of
 290 2^n . This is illustrated in Figure 2.4.

291 The results from Figure 2.5 show that the solutions converge as the grid size becomes refined.

292 2.4.3 Results

293 Here the main comparison that analysis the effects of using Algorithm 1 with different P values.
 294 Recall that the main observation is for $P = 1$, which corelates to the semi-implicit method. Along
 295 with accuracy, the simulation time will be tracked. This is because it represents another metric for
 296 which the viability of the fully-implicit method can be verified. Theoretically there should be a
 297 decrease in the error with the fully-implicit method as the value for P increases. Therefore, this needs
 298 to be weighted against the cost of computational intensity and the increase of the simulation time.

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

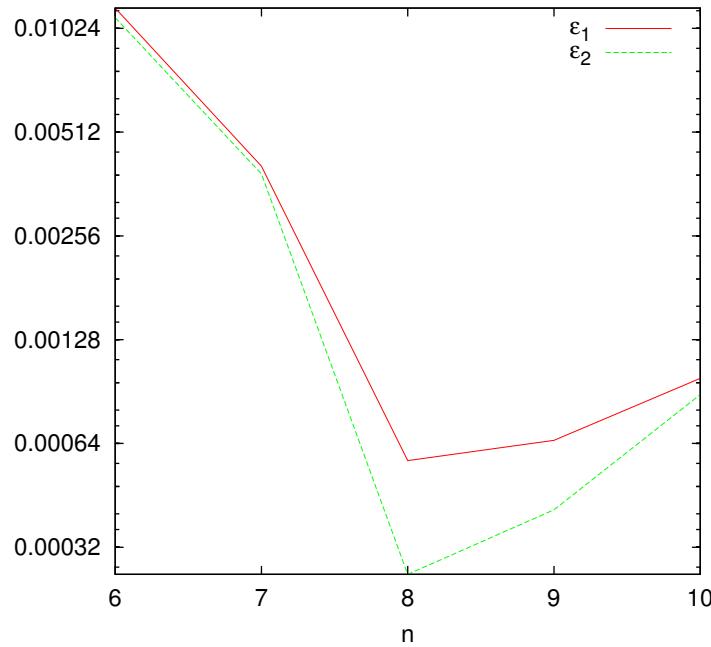


Figure 2.5: Plot showing the convergence of solutions based on changed in Δx . The computations are of ϵ_{ℓ_1} , ϵ_{ℓ_2} , and $\epsilon_{\ell\infty}$ with grid-size following $s(n) = 2^{n-1} + 1$.

P	Simulation Time	ϵ_{ℓ_1}	ϵ_{ℓ_2}	$\epsilon_{\ell\infty}$
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.

300 References

- 301 R. Barrett, M. Berry, Chan T.F., J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine,
302 and H. van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative*
303 *Methods*. Society for Industrial and Applied Mathematics, 1st edition, 1987.
- 304 J.C. Butcher. *Numerical Methods for Ordinary Differential Equations*. Wiley, 2nd edition, June 2008.
- 305 A Dumitrache. *Understanding Biofilms of Anaerobic, Thermophilic and Cellulolytic Bacteria: A*
306 *Study towards the Advancement of Consolidated Bioprocessing Strategies*. PhD thesis, University
307 of Toronto, 2014.
- 308 HJ Eberl and L Demaret. A finite difference scheme for a doubly degenerate diffusionreaction equa-
309 tion arising in microbial ecology. *Electron. J. Differential Equations*, page 7795, 2007.
- 310 HJ Eberl, DF Parker, and MCM van Loosdrecht. A new deterministic spatio-temporal continuum
311 model for biofilm development. *Journal of Theoretical Medicine*, pages 161–175, 2001.
- 312 S. Geršgorin. über die abgrenzung der eigenwerte einer matrix. *Bulletin de l'Académie des Sciences*
313 *de l'URSS. Classe des sciences mathématiques et na*, pages 749–754, 1931.
- 314 DR Noguera, G Pizarro, DA Stahl, and BE Rittmann. Simulation of multispecies biofilm development
315 in three dimensions. *Water Science and Technology*, 39:123–130, 1999.
- 316 BE Rittmann and PL McCarty. Model of steady-state biofilm kinetics. *Biotechnology and Bioengi-
317 neering*, 22:2343–2357, 1980.
- 318 Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematic,
319 2nd edition, 2003.

- 320 S Sirca and M Horvat. *Computational Methods for Physicistsl: Compendium for Students*. Springer,
321 2012.
- 322 Z-W Wang, S-H Lee, JG Elkins, and JL Morrell-Falvey. Spatial and temporal dynamics of cellulose
323 degradation and biofilm formation by caldicellulosiruptor obsidiansis and clostridium thermocel-
324 lum. *AMB Express*, 1:30, 2011.

