

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	21

²⁷ **Chapter 1**

²⁸ **Model Definition**

²⁹ **1.1 History**

³⁰ The tradition biofilm model has been continually developed over many iterations since 1980. Rittmann
³¹ and McCarty (1980) formulated the steady-state biofilm model, developed using the concept that
³² biofilm growth would be the result of a steady flux from substrate. Since then the model have evolved
³³ to include modelling three-dimensional growth of multispecies anaerobic biofilms (Noguera et al.
³⁴ (1999)) and spatially heterogeneous biofilm structures (Eberl et al. (2001)).

³⁵ The modelling of *Clostridium Thermocellum* is unique because this cellulolytic anaerobic bacteria
³⁶ does not generate an extracellular polymeric substance. This uncharacteristic behaviour means that
³⁷ the mathematical model based on the work of Eberl and Demaret (2007) cannot be used as is. They
³⁸ modelled the biomass density and nutrient concentrations as a two-PDE-coupled system. Recently,
³⁹ Wang et al. (2011), used a cellular automata based model for simulating the growth of *Clostridium*
⁴⁰ *Thermocellum*. From this, better results were thought to derive from a continuous differential equa-
⁴¹ tion based model. Here the spatial diffusion of the substrait concentration is removed to mimic the
⁴² carbon substrait that is consumed by *Clostridium Thermocellum*. This results in a PDE-ODE-coupled
⁴³ system. This is based on the work done by Dumitrache (2014), where this same coupling was used
⁴⁴ and formulated.

45 1.2 Model Description

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

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

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

51 where

$$52 d(M) = d \frac{M^\alpha}{(1 - M)^\beta} \quad (1.3)$$

$$53 f(C) = u \frac{C}{k + C} - n \quad (1.4)$$

$$55 g(C) = y \frac{C}{k + C} \quad (1.5)$$

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

63 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

64 1.3 Nondimensionalization

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

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

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

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

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

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

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

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

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

⁷⁰ Using these, (1.1) and (1.2) can be simplified and nondimensionalized into,

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

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

⁷¹ where,

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

⁷³ with only $\delta, \kappa, \nu, \gamma$ as model parameters.

74 1.4 Parameters

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

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

86 **Chapter 2**

87 **Numerical Methods**

88 **2.1 Discretization**

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

91
$$\frac{M^{k+1} - M^k}{\Delta t} = \nabla_x(D(M^{k+1})\nabla_x M^{k+1}) + F(C^{k+1})M^{k+1}, \quad (2.1)$$

92
$$\frac{C^{k+1} - C^k}{\Delta t} = \frac{h}{2}(G(C^{k+1})M^{k+1} + G(C^k)M^k). \quad (2.2)$$
93

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

97 For this system, the region of consideration will be a rectangular region, Ω . This region has Neumann
98 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,
99 according to the biology of our system, the substrate does not diffuse across the region. The spatial
100 discretization will be through the Finite Difference Method as described in Saad (2003). Here, a
101 uniform $n \times m$ grid is used to discretize Ω . This means that the distance between grid points are the
102 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

103 at each grid point using a five-point stencil. To index the grid point, i and j are used such that
 104 $M_{i,j}^k \approx M(t^k, x_{1_i}, x_{2_j})$. Because of the five-point stencil, the boundary gridpoints will depend on
 105 ghost grid points. This means that the equation to solve interior grid points will differ slightly from
 106 boundary points.

107 The resulting equation for interior points, where $i \in (1, 2, \dots, n - 1)$ and $j \in (1, 2, \dots, m - 1)$, is

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

108 where $\mathbb{A} = \{(0, \pm 1), (\pm 1, 0)\}$.

110 The resulting equation for boundary points, when $(i, j) \in \{0, n\} \times \{0, 1, \dots, m\}$ or $(i, j) \in \{0, 1, \dots, n\} \times$
 111 $\{0, m\}$, is

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

113 where \mathbb{B} depends on the boundary position; $\mathbb{B}_1 = \{(0, -1)\}$, $\mathbb{B}_2 = \{(-1, 0)\}$, $\mathbb{B}_3 = \{(1, 0)\}$,
 114 $\mathbb{B}_4 = \{(0, 1)\}$. For the corner points, where two different boundaries connect, the result is to use a
 115 cross-product between \mathbb{B} 's, for example $\mathbb{B} = \mathbb{B}_1 \times \mathbb{B}_2$

116 For both (2.3 - 2.4) the arithmetic mean of the diffusion function, D , is taken because of the steep
 117 gradiant at the interface. Taking the value of the $D(M_{i+\frac{s}{2},j+\frac{r}{2}}^{k+1})$ might result in a zero and thus it nullify
 118 the diffusion effect.

119 2.2 Solving Method

120 Now there exist equations for which C and M can be solved, (2.2) and (2.3 - 2.4) respectivly. Using
 121 C^k and $M_{i,j}^k$ as approximations of the solutions for (1.14) will allow the system to be solved by

122 computing C^{k+1} and $M_{i,j}^{k+1}$. However, there are complications with trying to get an explicit formula
 123 for $M_{i,j}^{k+1}$ from (2.3 - 2.4) because of the dependency on M in $D(M)$. To remedy this, a fixed point
 124 iteration is introduced. In a single time step, the solutions for M and C can be solved using the
 125 previous time step solution in the follow manner:

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

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

129 where $(p) \in (0, 1, \dots, P)$. Note, that the equation for $M_{i,j}^{(p+1)}$ shown in (2.5) refers to the interior
 130 points only. A similar change is done for the boundary points but is not shown due to its complexity.
 131 It is important to show explicitly that the purpose of the fixed point iteration is to link two distinct
 132 times with P solutions in between them, such that:

$$\begin{aligned} M^{(p=0)} &= M^k, & M^{(p=P)} &= M^{k+1}, \\ 133 \quad C^{(p=0)} &= C^k, & C^{(p=P)} &= C^{k+1}. \end{aligned} \quad (2.7)$$

134 In this fixed point format, given by (2.5 - 2.6), the equations can be rearrange and solved by conven-
 135 tional methods.

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

$$\begin{aligned} \frac{M_{i,j}^k}{\Delta t} &= \sum_{(i,j) \in \mathbb{A}} \left(\frac{D(M_{i+s,j+r}^{(p+1)}) + D(M_{i,j}^{(p+1)})}{2\Delta x^2} \cdot M_{i+s,j+r}^{(p+1)} \right) \\ 138 \quad &+ \left(\sum_{(i,j) \in \mathbb{A}} \left(\frac{D(M_{i+s,j+r}^{(p+1)}) + D(M_{i,j}^{(p+1)})}{2\Delta x^2} \right) - F(C_{i,j}^{(p)}) + \frac{1}{\Delta t} \right) M_{i,j}^{(p+1)}. \end{aligned} \quad (2.8)$$

¹³⁹ From (2.8), a five-diagonal matrix can be created defined as,

$$\text{140 } A = \begin{pmatrix} a_{i,j} & a_{i+1,j} & & a_{i,j+1} & & \\ a_{i-1,j} & \ddots & \ddots & & \ddots & \\ & \ddots & \ddots & \ddots & & \ddots \\ a_{i,j-1} & & a_{i-1,j} & a_{i,j} & a_{i+1,j} & a_{i,j+1} \\ & \ddots & & \ddots & \ddots & \ddots & \ddots \\ & & a_{i,j-1} & a_{i-1,j} & a_{i,j} & a_{i+1,j} & a_{i,j+1} \\ & & & \ddots & \ddots & \ddots & \ddots \\ & & & & \ddots & \ddots & \ddots & a_{i+1,j} \\ & & & & a_{i,j-1} & a_{i-1,j} & a_{i,j} & \end{pmatrix} \quad (2.9)$$

¹⁴¹ where each $a_{i,j}$ is the coefficient based on (2.8).

¹⁴² Solving (2.9) can be done by use of the Conjugate Gradient method provided that certain conditions
¹⁴³ are satisfied.

¹⁴⁴ **Proposition 2.2.1.** *The matrix A , (2.9), is positive definite and symmetric when $\frac{1}{F(C_{i,j}^{(p)})} < \Delta t$.*

¹⁴⁵ *Proof.* Matrix A is positive definite if all the eigenvalues are positive. Using the Circle theorem
¹⁴⁶ described by Geršgorin (1931), the eigenvalues can be shown to be positive if, independently on all
¹⁴⁷ rows, the sum of the off-diagonals values is less than the diagonal value. This can be verified. From
¹⁴⁸ (2.8) it can be said that,

$$\text{149 } \sum_{(i,j) \in \mathbb{A}} \left(\frac{D(M_{i+\frac{s}{2}, j+\frac{r}{2}}^{(p)})}{\Delta x^2} \right) < \left(\sum_{(i,j) \in \mathbb{A}} \left(\frac{D(M_{i+\frac{s}{2}, j+\frac{r}{2}}^{(p)})}{\Delta x^2} \right) - F(C_{i,j}^{(p)}) + \frac{1}{\Delta t} \right). \quad (2.10)$$

¹⁵⁰ This simplifies to,

$$\text{151 } F(C_{i,j}^{(p)}) < \frac{1}{\Delta t} \quad (2.11)$$

¹⁵² The symmetry of A can be trivially shown if one considers the formation of the diagonals. On a

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

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

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

168 For solving (2.6), the equation can be rearranged into a quadratic form, substituting in $G(C)$ from
 169 (1.16)

$$170 \quad (C^{(p+1)})^2 + \left(\kappa - C^k + \frac{h}{2} \gamma M^{(p+1)} + \frac{h \gamma C^k M^k}{2 \kappa + C^k} \right) C^{(p+1)} + \left(-\kappa C^k + \frac{h \gamma \kappa C^k M^k}{2 \kappa + C^k} \right) = 0. \quad (2.12)$$

171 Using the quadratic equation results in,

$$172 \quad C^{(p+1)} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (2.13)$$

¹⁷³ for which,

$$a = 1$$

¹⁷⁴

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

¹⁷⁵ To determine which branch of (2.13) to use, a physical situation is used. Specifically the case where
¹⁷⁶ there exist no biomass, $M = 0$. The expected outcome is that no substrate is consumed and thus
¹⁷⁷ the substrait concentration will remain constant as a function of t . When the equations in (2.14) are
¹⁷⁸ evaluated at $M = 0$, the result it,

¹⁷⁹

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

¹⁸⁰ which can be used to evaluate (2.13) as,

¹⁸¹

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

¹⁸² Now, if the positive branch is used the above equation evaluates to $C^{(p+1)} = C^k$. This means that
¹⁸³ between any two distinct times, the substrait concentration will remain constants, which was expected.
¹⁸⁴ To further this confirmation, the negative branch results in $C^{(p+1)} = -\kappa$, a non-postive substrate
¹⁸⁵ concentration, which is not physically relavent.

¹⁸⁶

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

¹⁸⁷ where a, b , and c are defined in (2.14).

¹⁸⁸ Now that computable solutions for M and C at a single time step have been found, an algorithm to
¹⁸⁹ solve for the next time step can be established. Algorithm 1 shows the organization 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)

190
191 (1.14). If $P = 1$ then only a single iteration of the algorithm is applied, which correlates to a semi-
192 implicit method would behave. This would result in a change similar to how the Gauss-Seidal method
193 changes the Jacobi method; the values used would no longer be updated in a single timestep when
194 $P = 1$.

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

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

197
198 This mapping allows the system to be easily stored in diagonal format, since (2.9) has five distinct
199 diagonals.

200 2.3 Computational Setup

201 The implementation of Algorithm 1 was done with Fortran.

202 All the computations were run on a custom built workstation with an Intel Xeon CPU E5-2650 (1.2
203 GHz, 20MB cache size) and 32 GB RAM under Red Hat Enterprise Linux Server release 6.5 (San-
204 tiago). Running the computations with OpenMP, took advantage of 6 out of the 16 processors of the

205 Intel Xeon CPU, each with 2 threads. The GNU Fortran compiler, version 4.4.7, was used for all
 206 computations; the compiler arguments were

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

208 **2.4 Numerical Results**

209 With a defined method and computational setup a variety of simulations can be run to observe the
 210 accuracy and behaviour of the method. An examination of a typical simulation will show if the ex-
 211 pected behaviour is observed, validating the method as functioning. A convergence analysis for the
 212 method can be done to confirm that solutions from different grid sizes approach a single solution as
 213 they become more precise. This convergence test will also show the thresholds for an accurate simu-
 214 lation result, to help reduce the computation times. With a well-established method, the comparison
 215 between semi- and fully-implicit methods can be done.

216 **2.4.1 Basic Simulations**

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

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

$$222 M = \begin{cases} -\left(\frac{h}{d^4}\right) x^4 + h & , \text{if } y \leq d \\ 0 & , \text{otherwise} \end{cases} \quad (2.19)$$

$$C = 1$$

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

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

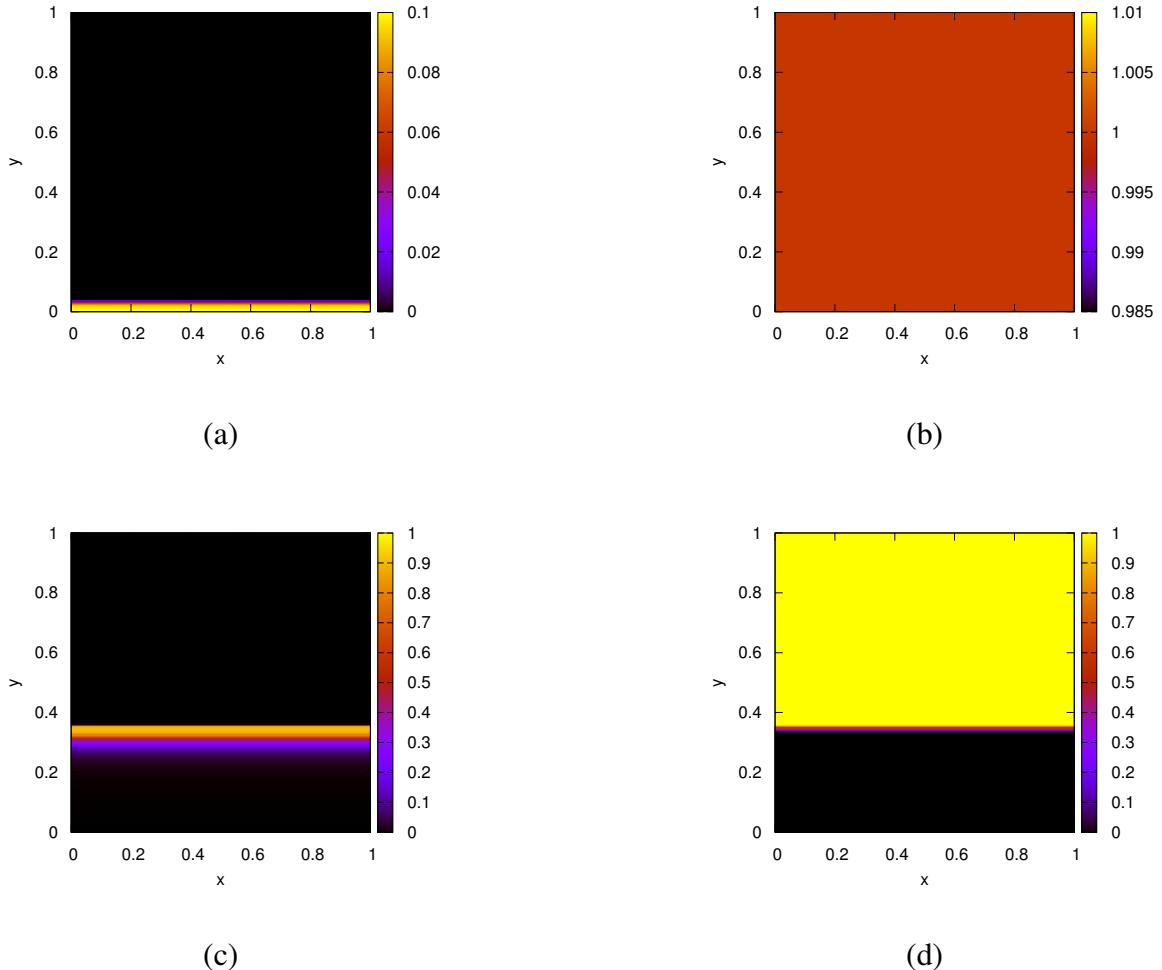


Figure 2.1: Solutions for (ac) M and (bd) C with 1D initial conditions defined in (2.19) at (ab) $t = 0$ and (cd) $t = 40$. Computed with a 1025×1025 grid and a timestep of $\Delta t = 10^{-3}$.

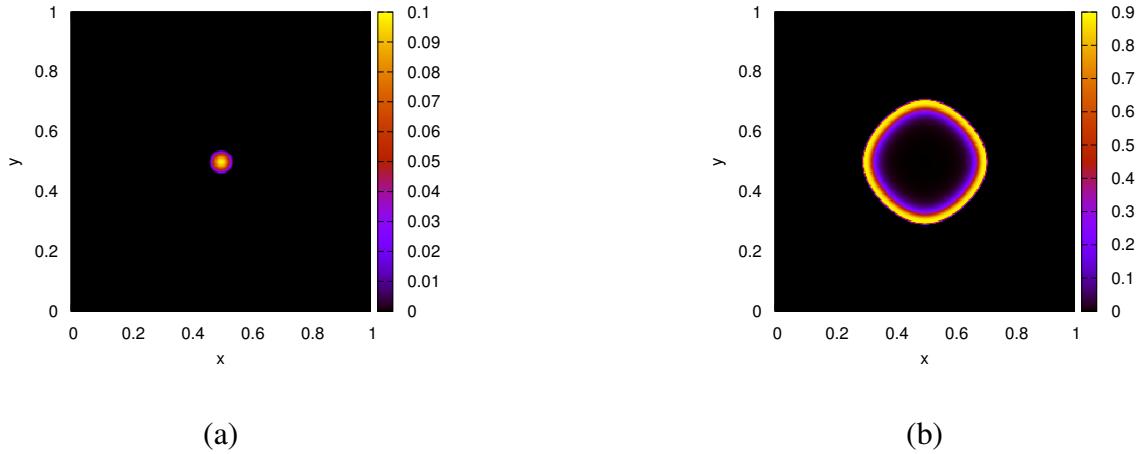


Figure 2.2: Solutions for M with spherical initial conditions defined by (2.20) at (a) $t = 0$ and (b) $t = 40$. Computed with a 1025×1025 grid and a timestep of $\Delta t = 10^{-3}$.

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

$$227 \quad M = \begin{cases} -\frac{h}{d^2} (x - 0.5)^2 + (y - 0.5)^2 + h & , \text{if } (x - 0.5)^2 + (y - 0.5)^2 < d^2 \\ 0 & , \text{otherwise} \end{cases}, \quad (2.20)$$

228 a test can be tried to see if the spherical nature of the solution is kept as time progresses. This can be
 229 seen in Figure 2.2, where at different times the shape of the biomass, M , is seen to remain spherical.

230 Both Figure 2.1 and Figure 2.2 increase the confidence that the spatial discretization did not introduce
 231 any loss of characteristics for the solutions.

232 The given boundary conditions and spatial discretization could be a possible source or sink of biomass.
 233 To ensure this is not the case, the total amount of biomass can be tracked. The total biomass cannot
 234 be exactly determined with the given growth rate function. This means that there will not be anything
 235 to measure the validity of the simulation solution against. However, if the growth rate were constant,
 236 the total biomass would be expected to grow exponentially like $y_0 e^{kt}$. This can be checked by tracking

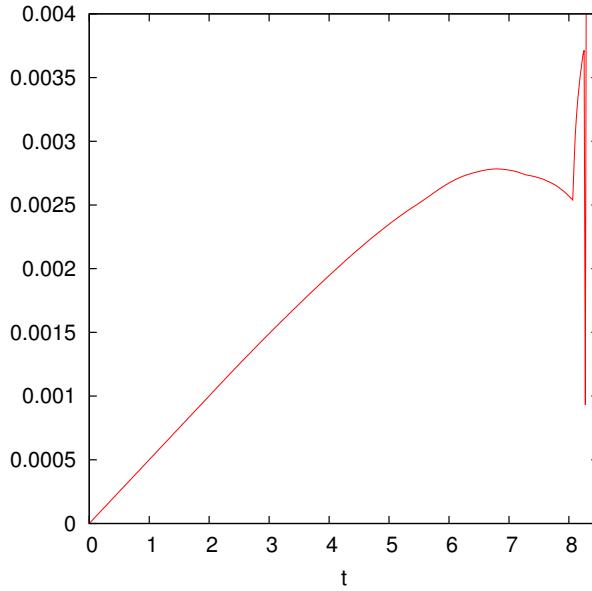


Figure 2.3: Plot of the relative error between the computed total biomass, $T_M(t)$, and the theoretical total biomass, $y_0 e^x$. The changes after $t = 8$ are from the biomass having completely filled the region Ω . This means that there is no physical space for the biomass to occupy and thus the growth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

²³⁷ the total biomass, now called $T_M(t)$, with the changed growth rate function:

$$\text{²³⁸ } F(C) = 1 \quad (2.21)$$

²³⁹ Tracking $T_M(t)$ can be done by,

$$\text{²⁴⁰ } T_M(t) = \int_{\Omega} M(t) dx. \quad (2.22)$$

²⁴¹ Numerically, this is computed by grid-wise summation,

$$\text{²⁴² } T_M(t^k) \approx T_M^k = \frac{\sum_i^n \sum_j^m M_{i,j}^k}{nm} \quad (2.23)$$

²⁴³ Figure 2.3 shows that the total biomass is as expected, since it matches the curve of $\frac{1}{3812.5} * e^k$, an
²⁴⁴ exponential function. The simulation results in Figure 2.3 are from the same as Figure 2.2 but at later
²⁴⁵ times so that the boundary effects can be observed.

²⁴⁶ **2.4.2 Convergence Analysis**

²⁴⁷ To validate the accuracy of the method, convergence analyses on the spatial discretizations will need
²⁴⁸ to be made. Then the comparison between the semi- and fully-implicit method established in Algo-
²⁴⁹ rithm 1 can investigated. First, a metric must be formed to enable consistent comparisons between
²⁵⁰ different simulation solutions. This metric will be referred to as the error.

²⁵¹ **2.4.2.1 Error Computations**

²⁵² The error is computed by taking the relative normed-difference between two solution in the following
²⁵³ fashion:

²⁵⁴

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

²⁵⁵ where u_1 represents one simulation solution and u_2 represents the solution that is theoretically more
²⁵⁶ accurate. The theoretical accuracy of u_2 derives from the fact that most comparisons will be done
²⁵⁷ between solutions that vary in only Δx or between semi- and fully- implicit. These are understood to
²⁵⁸ have the relation that a smaller Δx , and that the fully-implicit method is to be more accurate. There
²⁵⁹ is an assumption that both u_1 and u_2 have the same number of grid points, so that the difference can
²⁶⁰ be taken grid-wise.

²⁶¹ The results of the error computations, named ϵ_{sol} , is a numerical value for the difference between two
²⁶² solutions. This depends on the norm used during the computations. Here three norms will be used:

²⁶³

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

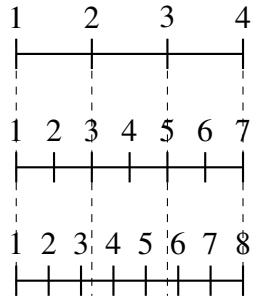
²⁶⁴

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

²⁶⁵

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

²⁶⁶ These different norms will all be used to create a broader understanding of the error. This creates three
²⁶⁷



4 grid points initially

(a) 7 grid points following $s(n) = 2^n + 1$ (b) 8 grid points following 2^{n+1}

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.

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

270 2.4.2.2 Grid Size Convergence

271 To observe the validity of the method, a test on the convergence of solutions based on the spatial
272 discretization is done. This will involve using the same simulation described in (2.19) due to the
273 simplicity.

274 The convergence will be tracked with only two forms of ϵ_{sol} ; ϵ_1 and ϵ_2 . This is because the value of ϵ_∞
275 doesn't vary with the grid size; the wave front is too sharp and with a change in grid size it will always
276 be the point of greatest error. Since there exist a difference in the number of grid points between the
277 different solutions, u_1 and u_2 , only the grid points in the more coarse refinement will be used. This
278 places a limitation on the selection of grid-sizes and so they will follow the series $s(n) = 2^{n-1} + 1$
279 for $n \in \mathbb{Z}$. So that the grid points can line up, without any use of linear interpolation, grid refine must
280 be using $s(n)$ instead of the typical grid doubling of 2^n . This is illustrated in Figure 2.4.

281 The results from Figure 2.5 show that the solutions converge as the grid size become refined with the
282 grid size.

283 2.4.3 Results

284 Here the main comparison that analyses the effects of using Algorithm 1 with different tolerances.
285 Recall that the main observation is for $tol. = 1$, which correlates to the semi-implicit method since it

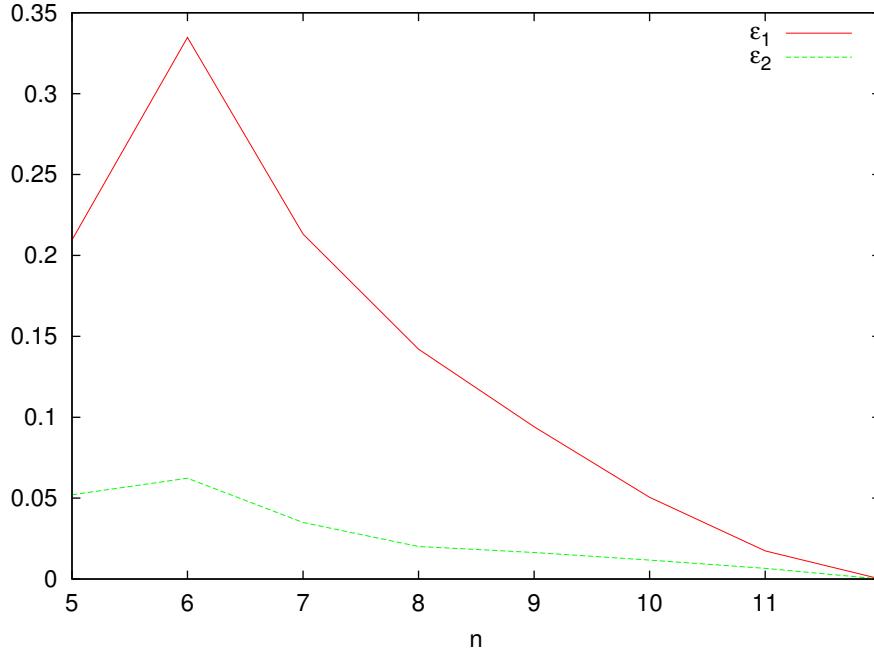


Figure 2.5: Plot showing the convergence of solutions based on changes in Δx . The computations are of ϵ_{ℓ_1} and ϵ_{ℓ_2} with grid-size following $s(n) = 2^{n-1} + 1$.

286 will allow only a single iteration of the algorithm.

287 The simulation used is the same as described in (2.1). The comparison will be on multiple metrics:

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

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

293 The value of ϵ_1 and ϵ_2 act as a measure of accuracy. Here, these values correspond to the difference be-
294 tween a pair of solutions, u_1 and u_2 . The (u_1, u_2) pairs are: $(1, 10^{-8})$, $(10^{-8}, 10^{-10})$, $(10^{-10}, 10^{-12})$, $(10^{-12}, 10^{-14})$.

295 Each row of Table 2.1 refers to the u_1 values of the pairs. Each difference was taken at the last
296 timestep.

297 Along with accuracy, the simulation time is tracked. This is because it represents another metric
298 for which the viability of the fully-implicit method can be verified. Theoretically there should be a

299 decrease in the error with the fully-implicit method as the value for *tol* decreases. Therefore, this
 300 needs to be weighted against the cost of computational intensity and the increase of the simulation
 301 time.

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

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

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			173.8489999999999	0.460937500000

Table 2.1: Results from running simulations with different Tol.

306

307 **References**

- 308 R. Barrett, M. Berry, Chan T.F., J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine,
309 and H. van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative*
310 *Methods*. Society for Industrial and Applied Mathematics, 1st edition, 1987.
- 311 J.C. Butcher. *Numerical Methods for Ordinary Differential Equations*. Wiley, 2nd edition, June 2008.
- 312 A Dumitrache. *Understanding Biofilms of Anaerobic, Thermophilic and Cellulolytic Bacteria: A*
313 *Study towards the Advancement of Consolidated Bioprocessing Strategies*. PhD thesis, University
314 of Toronto, 2014.
- 315 HJ Eberl and L Demaret. A finite difference scheme for a doubly degenerate diffusionreaction equa-
316 tion arising in microbial ecology. *Electron. J. Differential Equations*, page 7795, 2007.
- 317 HJ Eberl, DF Parker, and MCM van Loosdrecht. A new deterministic spatio-temporal continuum
318 model for biofilm development. *Journal of Theoretical Medicine*, pages 161–175, 2001.
- 319 S. Geršgorin. über die abgrenzung der eigenwerte einer matrix. *Bulletin de l'Académie des Sciences*
320 *de l'URSS. Classe des sciences mathématiques et na*, pages 749–754, 1931.
- 321 DR Noguera, G Pizarro, DA Stahl, and BE Rittmann. Simulation of multispecies biofilm development
322 in three dimensions. *Water Science and Technology*, 39:123–130, 1999.
- 323 BE Rittmann and PL McCarty. Model of steady-state biofilm kinetics. *Biotechnology and Bioengi-
324 neering*, 22:2343–2357, 1980.
- 325 Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematic,
326 2nd edition, 2003.

- 327 S Sirca and M Horvat. *Computational Methods for Physicistsl: Compendium for Students*. Springer,
328 2012.
- 329 Z-W Wang, S-H Lee, JG Elkins, and JL Morrell-Falvey. Spatial and temporal dynamics of cellulose
330 degradation and biofilm formation by caldicellulosiruptor obsidiansis and clostridium thermocel-
331 lum. *AMB Express*, 1:30, 2011.

