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

Chapter 1

Introductions

In here will be a bunch of info on the literature of biofilm simulations, and probably should mention the cellular automata model of C.Thermocellum in here and how this is mainly trying to make a continuous model of that simulation.

Add also how that the existance of a travelling wave solution will be looked for, but not proven.... and how we will also try to model the growth of C.thermocellum based on the production of CO2.

Chapter 2

Numerics

2.1 Model

```
Talk about: the origins of the model (Hermann?)
            the origins for the functions D(M), F(C,M) and G(C,M).
            the parameters and where their values come from.
            More....?
```

The model used for simulations is based on the deterministic model developed in Eberl et al. (2001) to model 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 to grow. The PDE-ODE-coupled system is purposed as,

$$M_\tau = \nabla_\chi \left(\overline{D}(M) \nabla_\chi M \right) + \overline{f}(\overline{C}, M) M \tag{2.1}$$

$$\overline{C}_\tau = -\overline{g}(\overline{C}) M \tag{2.2}$$

where

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

$$\overline{f}(\overline{C}, M) = \frac{\overline{\mu}\overline{C}}{\overline{k} + \overline{C}} \left(1 - \left(\frac{M}{\overline{C}} M_0\right)^{\gamma}\right) \tag{2.4}$$

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

Here in (2.1) the spatial diffusion of biomass is density dependent diffusion, given by (2.3), and the growth rate of biomass, $\overline{f}(\overline{C}, M)$, is given by (2.4). The growth rate is Monod kinetic growth multiplied by logistic growth to enforce a carrying capacity and death rate in the biomass. In (2.2) there is only a consumption term from the bacteria consuming the carbon substrait.

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

Variable/Parameter	Dimensions	Parameter Value
τ	$[days]$	$-$
χ	$[meters]$	$-$
M	$[-]$	$-$
C	$[\frac{grams}{meters^3}]$	$-$
$\overline{\delta}$	$[\frac{meters^2}{days}]$	10^{-12}
α	$[-]$	4
β	$[-]$	4
$\overline{\mu}$	$[days^{-1}]$	6
$\overline{k}$	$[\frac{grams}{meters^3}]$	4
γ	$[-]$	0.5
$\overline{\nu}$	$[\frac{grams}{meters^3 \cdot days}]$	$\frac{\mu M_0}{0.63}$
M_0	$[-]$	10000
C_0	$[\frac{grams}{meters^3}]$	30

Table 2.1: List of parameters and their dimensions

2.2 Nondimensionalization

For this system, it is better to nondimensionalize. To do this, the following variable changes are used:

$$x = \frac{\chi}{L} \implies Ldx = d\chi \tag{2.6}$$

$$t = \mu\tau \implies \frac{1}{\mu}dt = d\tau \tag{2.7}$$

$$C = \frac{\overline{C}}{C_0} \tag{2.8}$$

$$\delta = \frac{1}{\mu L^2}\overline{\delta} \tag{2.9}$$

$$k = \frac{\overline{k}}{C_0} \tag{2.10}$$

$$\nu = \frac{\overline{\nu}}{\mu C_0} \tag{2.11}$$

$$\mu = \overline{\mu} \tag{2.12}$$

This gives the system

$$M_t = \frac{1}{\mu L^2}\nabla_x\left(\hat{D}(m)\nabla_x M\right) + \frac{1}{\mu}\hat{f}(C, M) \tag{2.13}$$

$$C_t = \frac{-1}{C_0\mu}\hat{g}(C, M) \tag{2.14}$$

where

$$\hat{D}(M) = \mu L^2\delta\frac{M^\alpha}{(1-M)^\beta} \tag{2.15}$$

$$\hat{f}(C, M) = \frac{\mu C C_0}{kC_0 + CC_0}M\left(1 - \left(\frac{M}{C}\frac{M_0}{C_0}\right)^\gamma\right) \tag{2.16}$$

$$\hat{g}(C, M) = -\frac{\mu C_0\nu C C_0}{kC_0 + CC_0}M \tag{2.17}$$

This can be greatly simplified by cancelling out parameters.

$$M_t = \nabla_x \left(\delta \frac{M^\alpha}{(1-M)^\beta} \nabla_x M \right) + \frac{C}{k+C} M \left(1 - \left(\frac{M}{C} \frac{M_0}{C_0} \right)^\gamma \right) \tag{2.18}$$

$$C_t = -\frac{\nu C}{k+C} M \tag{2.19}$$

Now we can name functions and get the final nondimensionalized system.

$$M_t = \nabla_x \left(D(M) \nabla_x M \right) + f(C, M) \tag{2.20}$$

$$C_t = -g(C, M) \tag{2.21}$$

where

$$\begin{aligned} D(M) &= \delta \frac{M^\alpha}{(1-M)^\beta} \\ f(C, M) &= \frac{C}{k+C} M \left(1 - \left(\frac{M}{C} \frac{M_0}{C_0} \right)^\gamma \right) \\ g(C, M) &= \frac{\nu C}{k+C} M \end{aligned} \tag{2.22}$$

with parameter values

$$
\begin{aligned}
L &= 0.01m \\
C_0 &= 30\frac{g}{m^3} \\
M_0 &= 1.000 \times 10^4 \frac{g}{m^3} \\
\delta &= 1.667 \times 10^{-9} \\
k &= 0.1333 \times 10^{-1} \\
\nu &= 5.291 \times 10^2
\end{aligned}
\tag{2.23}
$$

2.3 Methods

Describe the method of solveing. Use MATH6020 project description as a baseline (since its mainly the same method). Should give enough detail so that this method can be replicated (verbatim psuedo code?). essentially describing the discritization of M so that it can be solved using finite difference method and how C is solved by trapezidral rule.

2.3.1 Substrate Concentration

The substrate concentration is represented by the following equation:

$$C_t = g(C, M). \tag{2.24}$$

For our case we will use

$$g(C, M) = -\frac{\nu C}{k + C} M. \tag{2.25}$$

To solve for C we first apply the Fundemental Theorem of Calculus and the Trapizoidal Rule to (2.24).

$$
\begin{aligned}
\int_{t_n}^{t_{n+1}} C_t dt &= \int_{t_n}^{t_{n+1}} g(C, M) dt \\
C_{t_{n+1}} - C_{t_n} &= \frac{(t_{n+1} - t_n)}{2} \left(g(C_{t_{n+1}}, M_{t_{n+1}}) + g(C_{t_n}, M_{t_n}) \right).
\end{aligned}
\tag{2.26}
$$

For convience we will henceforth let $C_n = C_{t_n}$, $M_n = M_{t_n}$, and $h = t_{n+1} - t_n$. Now we substituite (2.25) and try to solve explicitly for C_{n+1}

$$\begin{aligned} C_{n+1} - C_n &= \frac{h}{2}\left(-\frac{\nu C_{n+1}}{k + C_{n+1}} M_{n+1} - \frac{\nu C_n}{k + C_n} M_n\right) \\ C_{n+1}(k + C_{n+1}) - C_n(k + C_{n+1}) &= -\frac{h}{2}\nu M_{n+1} C_{n+1} - \frac{h}{2}\frac{\nu C_n M_n}{k + C_n}(k + C_{n+1}) \end{aligned} \tag{2.27}$$

This gives us the following quadratic equation

$$C_{n+1}^2 + \left(k - C_n + \frac{h}{2}\nu M_{n+1} + \frac{h}{2}\frac{\nu C_n M_n}{k + C_n}\right) C_{n+1} + \left(-kC_n + \frac{h}{2}\frac{\nu k C_n M_n}{k + C_n}\right) = 0 \tag{2.28}$$

from which we can identify

$$\begin{aligned} a &= 1 \\ b &= k - C_n + \frac{h}{2}\nu M_{n+1} + \frac{h}{2}\frac{\nu C_n M_n}{k + C_n} \\ c &= -kC_n + \frac{h}{2}\frac{\nu k C_n M_n}{k + C_n} \end{aligned} \tag{2.29}$$

We can now solve for C_{n+1} using the quadratic formula

$$C_{n+1} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{2.30}$$

To figure out which value of C_{n+1} we use, we test the physical situation in which there are no bacteria consuming the substrate, that is $M = 0$. We expect that the substrate level will remain constant from one timestep to the next which this condition. So when $M = 0$ we have that,

$$a = 1, \quad b = k - C_n, \quad c = -kC_n, \tag{2.31}$$

which we can use with (2.30) to get

$$\begin{aligned} C_{n+1} &= \frac{-(k - C_n) \pm \sqrt{(k - C_n)^2 - 4(-kC_n)}}{2} \\ &= \frac{1}{2}\left(C_n - k \pm \sqrt{k^2 + 2C_n + C_n^2}\right) \\ &= \frac{1}{2}\left(C_n - k \pm (k + C_n)\right). \end{aligned} \tag{2.32}$$

Now, if we use the $'+'$ then $C_{n+1} = C_n$ and if we use the $'-'$ then $C_{n+1} = -k$. Since we cannot have a negative substrate concentration, and we also expected the substrate level to remain constant with these conditions, we can conclude that

$$C_{n+1} = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \tag{2.33}$$

where a, b, and c are defined in (2.29)

2.3.2 Biomass Density

We need to discritize the following equation, where $M(x, y, t) \equiv M$,

$$M_t = \nabla(D(M)\nabla M) + f(C, M)M \tag{2.34}$$

with respect to both time and space.

First we expand $\nabla(D(M)\nabla M)$,

$$M_t = \frac{\partial}{\partial x}\left(D(M)\frac{\partial}{\partial x}M\right) + \frac{\partial}{\partial y}\left(D(M)\frac{\partial}{\partial y}M\right) + f(C, M)M. \tag{2.35}$$

Now we can discritize for space, (Note: $D_{i,j} = D(M(x_i, y_j))$)

$$M_t = \frac{1}{\Delta x^2}\left[D_{i+\frac{1}{2},j}\left(M_{i+1,j} - M_{i,j}\right) - D_{i-\frac{1}{2},j}\left(M_{i,j} - M_{i-1,j}\right)\right] + \frac{1}{\Delta y^2}\left[D_{i,j+\frac{1}{2}}\left(M_{i,j+1} - M_{i,j}\right) - D_{i,j-\frac{1}{2}}\left(M_{i,j} - M_{i,j-1}\right)\right] + f(C, M)M, \tag{2.36}$$

and also for time,

$$\frac{M^{n+1} - M^n}{\Delta t} = \frac{1}{\Delta x^2}\left[D_{i+\frac{1}{2},j}\left(M^{n+1}_{i+1,j} - M^{n+1}_{i,j}\right) - D_{i-\frac{1}{2},j}\left(M^{n+1}_{i,j} - M^{n+1}_{i-1,j}\right)\right] \tag{2.37}$$

$$+ \frac{1}{\Delta y^2}\left[D_{i,j+\frac{1}{2}}\left(M^{n+1}_{i,j+1} - M^{n+1}_{i,j}\right) - D_{i,j-\frac{1}{2}}\left(M^{n+1}_{i,j} - M^{n+1}_{i,j-1}\right)\right] + f(C^n, M^n)M^{n+1}. \tag{2.38}$$

We want to solve this using a linear solver so we rearrange the equation in a form that separates the $M^(n+1)$ terms by their i, j-components

$$\begin{aligned}\frac{-M^n}{\Delta t} = {} & \frac{D_{i,j-\frac{1}{2}}}{\Delta y^2} \cdot M^{n+1}_{i,j-1} + \frac{D_{i-\frac{1}{2},j}}{\Delta x^2} \cdot M^{n+1}_{i-1,j} \\ & + \left[-\frac{D_{i,j-\frac{1}{2}}}{\Delta y^2} - \frac{D_{i-\frac{1}{2},j}}{\Delta x^2} - \frac{D_{i+\frac{1}{2},j}}{\Delta x^2} - \frac{D_{i,j+\frac{1}{2}}}{\Delta y^2} + f(C^n, M^n) - \frac{1}{\Delta t}\right] \cdot M^{n+1}_{i,j} \\ & + \frac{D_{i+\frac{1}{2},j}}{\Delta x^2} \cdot M^{n+1}_{i+1,j} + \frac{D_{i,j+\frac{1}{2}}}{\Delta y^2} \cdot M^{n+1}_{i,j+1} \\ & + f(C^n, M^n)M^{n+1}\end{aligned} \tag{2.39}$$

Lastly, multiply both sides by -1 for postive definitness

$$\begin{aligned}\frac{M^n}{\Delta t} = &\frac{-D_{i,j-\frac{1}{2}}}{\Delta y^2}\cdot M^{n+1}_{i,j-1} + \frac{-D_{i-\frac{1}{2},j}}{\Delta x^2}\cdot M^{n+1}_{i-1,j} \\ &+ \left[\frac{D_{i,j-\frac{1}{2}}}{\Delta y^2} + \frac{D_{i-\frac{1}{2},j}}{\Delta x^2} + \frac{D_{i+\frac{1}{2},j}}{\Delta x^2} + \frac{D_{i,j+\frac{1}{2}}}{\Delta y^2} - f(C^n, M^n) + \frac{1}{\Delta t}\right]\cdot M^{n+1}_{i,j} \\ &+ \frac{-D_{i+\frac{1}{2},j}}{\Delta x^2}\cdot M^{n+1}_{i+1,j} + \frac{-D_{i,j+\frac{1}{2}}}{\Delta y^2}\cdot M^{n+1}_{i,j+1}\end{aligned}$$

2.3.3 Finite Difference Method

b) *Discretize equation (**??**) on a uniform rectangular grid with step-size $\Delta x = \Delta y = h = 1/n$ on a grid size $n \times m$. Note that this implies $H = m/n$. The gridpoints are (x_i, y_i) with $x_i = ih,\ i = 0, 1, \ldots, n$ and $y_j = jh,\ j = 0, 1, \ldots, m$. To this end, discretize equation (**??**) to obtain a sparse linear system of the form*

$$Ap = b, \tag{2.40}$$

where the matrix A can be stored in sparse diagonal format.

Answer:

To discretize equation (**??**) we first need to create our grid. For this problem we use an orthogonal uniform grid for simplicity. From this we get that our grid points are (x_i, y_j), with $x_i = \frac{i}{n},\ i = 0, 1, \ldots, n$ and $y_j = \frac{j}{m},\ j = 0, 1, \ldots, m$. We can approximate (**??**) at a grid point (i, j) as

$$\begin{aligned}\frac{1}{\Delta t^2}\big[&k_{i,j-1/2}p_{i,j-1} + k_{i-1/2,j}p_{i-1,j} \\ &- \left(k_{i,j-1/2} + k_{i-1/2,j} + k_{i+1/2,j} + k_{i,j+1/2}\right)p_{i,j} \\ &+ k_{i+1/2,j}p_{i+1,j} + k_{i,j+1/2}p_{i,j+1}\big] = k_{i,j}p_{i,j},\end{aligned} \tag{2.41}$$

where $k_{i\pm1/2,j\pm1/2} = k(x_i \pm \frac{h}{2}, y_j \pm \frac{h}{2})$.

This means that at each grid point (i, j), we have dependency on $p_{i,j}, p_{i\pm1,j}$, and $p_{i,j\pm1}$. This results

in a system of $N = nm$ linear equations for N unknown $p_{i,j}$.

Each interior point can be computed using (2.41). For the Boundary points we take special considerations. At $x = 0$ and $x = 1$ we have Dirichlet Boundary Conditions. These grid points are excluded from the matrix computations since their values are known. At $y = 0$ and $y = 1$ we have Neumann Boundary Conditions. For these a second order approximation of the derivative is used.

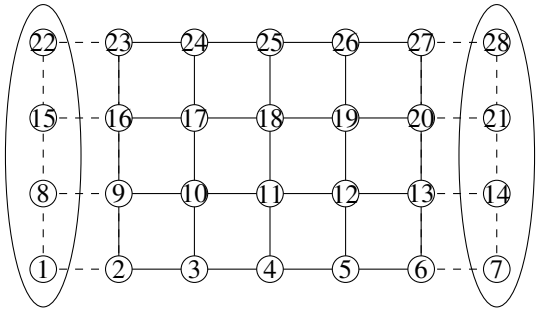


Figure 2.1: An example of the grid ordering on an 7 x 4 grid.

To solve this system, the problem is converted into the form

$$\mathcal{A}p = b \tag{2.42}$$

where $\mathcal{A}$ is the coefficents for each grid point, p is the solution vector, and b is the boundary conditions. To compute this a bijective mapping to convert the 2D grid into a 1D array is required, the following mapping is used here,

$$\begin{aligned} \pi : \{0, \ldots, n\} \times \{0, \ldots, m\} &\to \{1, \ldots, (n+1)(m+1)\} \\ (i, j) \qquad &\mapsto \qquad \pi(i, j) \end{aligned} \tag{2.43}$$

An example of this grid ordering can be seen in Figure 2.1.

The matrix is stored in diagonal format and since there are five unknowns for each linear equation the matrix will be banded with five diagonals. The numbering of these diagonals in the matrix can be seen in Figure 2.2

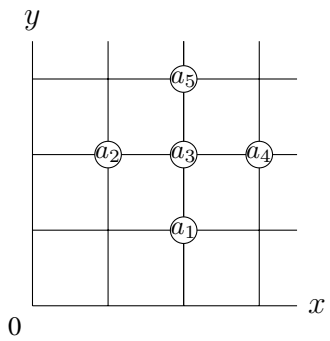
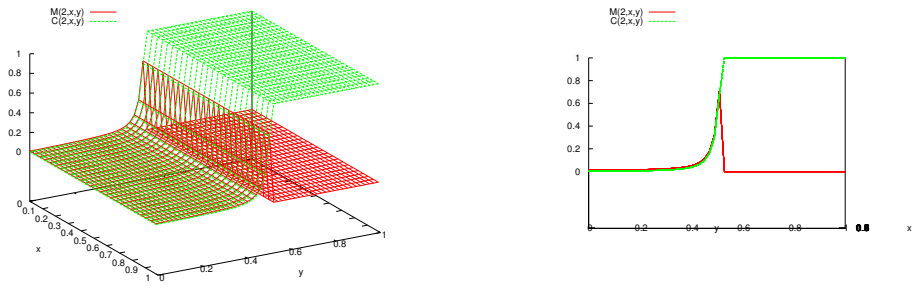


Figure 2.2: A visual of the grid point dependency and their numbering for the diagonally formatted matrix

2.4 Reduce to 1D problem

The system (2.20 - 2.22) can be reduced to a $1D$ problem.

Since the initial conditions are homogenous with respect to x, the x axis can be ignored and only the y-z axis is needed for visualization. Looking at Figure 2.3, the homoginity is clear.



(a) (b)

Figure 2.3: Graph of (a) 3D view of $M(t, x, y)$ and $C(t, x, y)$, (b) Side profile view of $M(t, x, y)$ and $C(t, x, y)$ at $t = 2$.

This can be shown quantitatily by comparing the maximum and minimum values of M and C with respect to the x-axis (Figure 2.4).

The difference between the max and min remains constant with respect to time (Table 2.2), this was

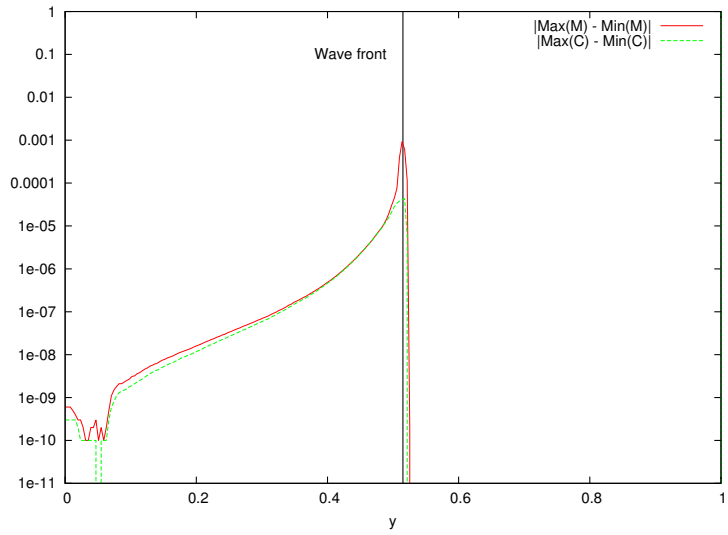


Figure 2.4: Graph of difference between max and min of M and difference between max and min of C.

t	δ_M	δ_C
0.0	0.0	0.0037109375
0.5	1.71577851563e-06	0.00371116503633
1.0	7.62408671875e-06	0.00371156747773
1.5	9.04633828125e-06	0.00371216829336
2.0	7.648728125e-06	0.00371311224609
2.5	2.58707890625e-06	0.00371457725742
3.0	0.00162623974609	0.00240237613672
3.5	5.35270417969e-05	7.24265101562e-05
4.5	1.07250539062e-05	2.12679726563e-05
5.0	1.09254414063e-06	9.720734375e-06

Table 2.2: A table of values for $\delta_M(t)$ and $\delta_C(t)$

calculated by,

$$\begin{aligned}\delta_M(t) &= \frac{1}{256}\sum_{i=1}^{n}\left(\sup_{j\in(1,m)} M(x_i,y_j) - \inf_{k\in(1,m)} M(x_i,y_k)\right)\\ \delta_C(t) &= \frac{1}{256}\sum_{i=1}^{n}\left(\sup_{j\in(1,m)} C(x_i,y_j) - \inf_{k\in(1,m)} C(x_i,y_k)\right)\end{aligned} \tag{2.44}$$

So by taking the average of the points along the x-axis we can get a 2D plot as seen in Figure 2.5.

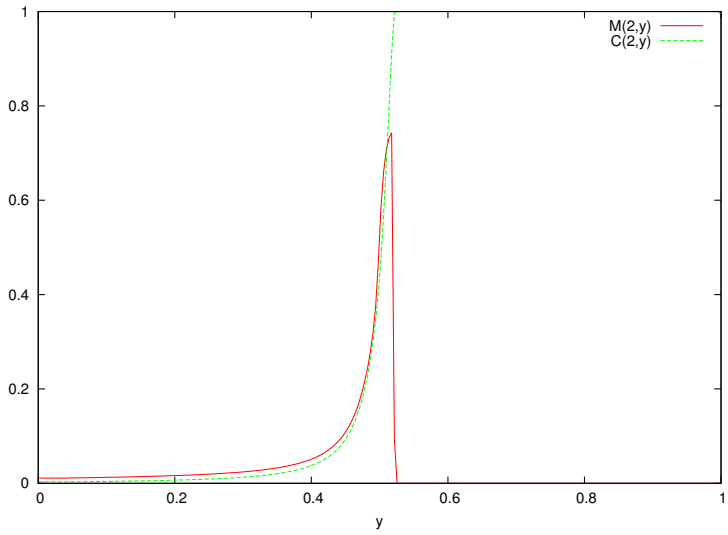


Figure 2.5: Graph of M(2,y) and C(2,y), now reduced to a 2D plot.

2.5 Convergence Results

The accuracy of this method depends on the grid size used. A convergence analysis is done to observe at what resolution sufficient accuracy is seen. This is done by comparing the relative error in average M between two solutions with different Δx. We call this difference δ, which we compute as

$$\delta_k = \frac{\left|\overline{M}_{k+1} - \overline{M}_k\right|}{\overline{M}_{k+1}} \tag{2.45}$$

where $\overline{M}_k(t)$ is the average biomass at time t, solved with $\Delta x = 2^{-k}$. Figure 2.6 was computed using (2.45) with $k = 7, 8, ..., 14$.

This convergence analysis shows a relative error less then 0.02% can be achieved by using $\Delta x = 2^{-13}$. This is considered an sufficient accuracy.

A second convergence result can be done by instead comparing each gridpoint between solutions. If we define

$$\sigma_k(t) = 2^{-k} \sum_{i,j} |M^{k+1}(t, x_i, y_j) - M^k(t, x_i, y_i)|, \tag{2.46}$$

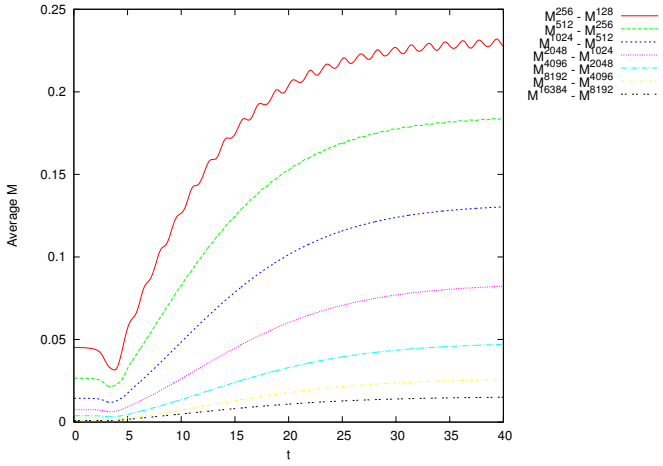


Figure 2.6: Comparison of mean M, using (2.45) with $i = 7, 8, ..., 14$

and

$$\rho_k = \frac{1}{n(T)} \sum_{\forall t \in T} \sigma_k(t). \tag{2.47}$$

In (2.46), $M^k(t, x_i, y_i)$ refers to the biomass when solved with $\Delta x = 2^{-k}$ at time t and gridpoint (x_i, y_i). The value of $\sigma_k(t)$ is the average difference between related gridpoints of solutions solved with different Δx values at a specific time t. In (2.47), $n(T)$ refers to the cardinality of T, which is the number of outputted times we have. The value of ρ is the average difference across all $t \in T$.

Figure 2.7: Graph of ρ_k, for $k = 7, 8, ..., 13$ and $T = 0, 2, ..., 40$

This can be extended to check convergence in Δt by defining

$$\overline{\rho}_\tau = \sum_k \rho_k^\tau \tag{2.48}$$

where ρ_k^τ is the same computations for (2.47) done with $\Delta t = \tau$.

2.6 Iterations

In here will be the description of the method and results of iterating between solvings of M and C until the difference has convergenced to a a specificed epsilon. Also mention in here the effects it has

on the solutions (not much) and computation time (2x longer ish?). Also discuss the single iterations and its effects (solve M -¿ solve C -¿ solve M again).

The system

$$M_t = \nabla_x \left(D(M) \nabla_x M \right) + f(C, M) M \tag{2.49}$$

$$C_t = -g(C, M) \tag{2.50}$$

where

$$D(M) = \delta \frac{M^\alpha}{(1 - M)^\beta} \tag{2.51}$$

$$f(C, M) = \frac{C}{k + C} \left(1 - \left(\frac{M M_0}{C C_0 + \epsilon} \right)^\gamma \right) \tag{2.52}$$

$$g(C, M) = \frac{\nu C}{k + C} M \tag{2.53}$$

is solved on a rectangular region with length L and width λL with the following parameter values,

$$\begin{aligned} L &= 0.01 \\ \lambda &= \frac{1}{128} & C_0 &= 30 \\ \epsilon &= 10^{-8} & M_0 &= 30 \\ \alpha &= 4 & \delta &= \frac{10^{-7}}{\mu L^2} \approx 10^-4 \\ \beta &= 4 & k &= \frac{4}{C_0} \approx 0.1333 \\ \gamma &= 0.5 & \nu &= \frac{M_0}{0.63 C_0} \approx 1.59, \\ \mu &= 6 \end{aligned} \tag{2.54}$$

and with initial conditions

$$C = 1$$

$$M = \begin{cases} -(\frac{h}{d^4}) x^4 + h & \text{if } x < 0.04 \\ 0 & \text{otherwise} \end{cases} \tag{2.55}$$

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

Using simulation code version 0.4 the effect of iterating between solving M and C was examined. Algorithm 1 shows how the iterations was implemented. The following experiments were run on the simulator.

Data: eSoln = 10^{-12}; M_{prev} and C_{prev} is previous timestep solutions
begin
  while *diffC* + *diffM* $>$ *eSoln* **do**
    Solve for M^{i+1} using C^i and M_{prev}
    Solve for C^{i+1} using C_{prev}, M^{i+1}, and M_{prev}
    Let diffC $= (C^{i+1} - C^i)$
    Let diffM $= (M^{i+1} - M^i)$
    Let $C^i = C^{i+1}$
    Let $M^i = M^{i+1}$
  end
end

Algorithm 1: Algorithm for iterating between solutions.

Figure 2.8 shows how the error between the iterated solutions changes the solutions. Notice that there is very little difference.

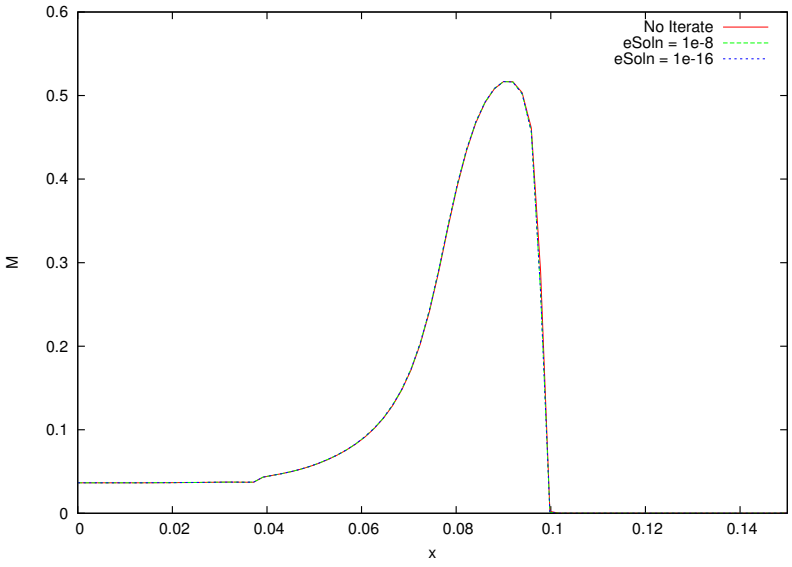


Figure 2.8: Convergence of solutions for no iterations between solutions, or iterating with different accuracy values.

Figure 2.9 shows, for different Δt values, the convergence of solutions as Δx becomes smaller. Very minute differences between Δt values.

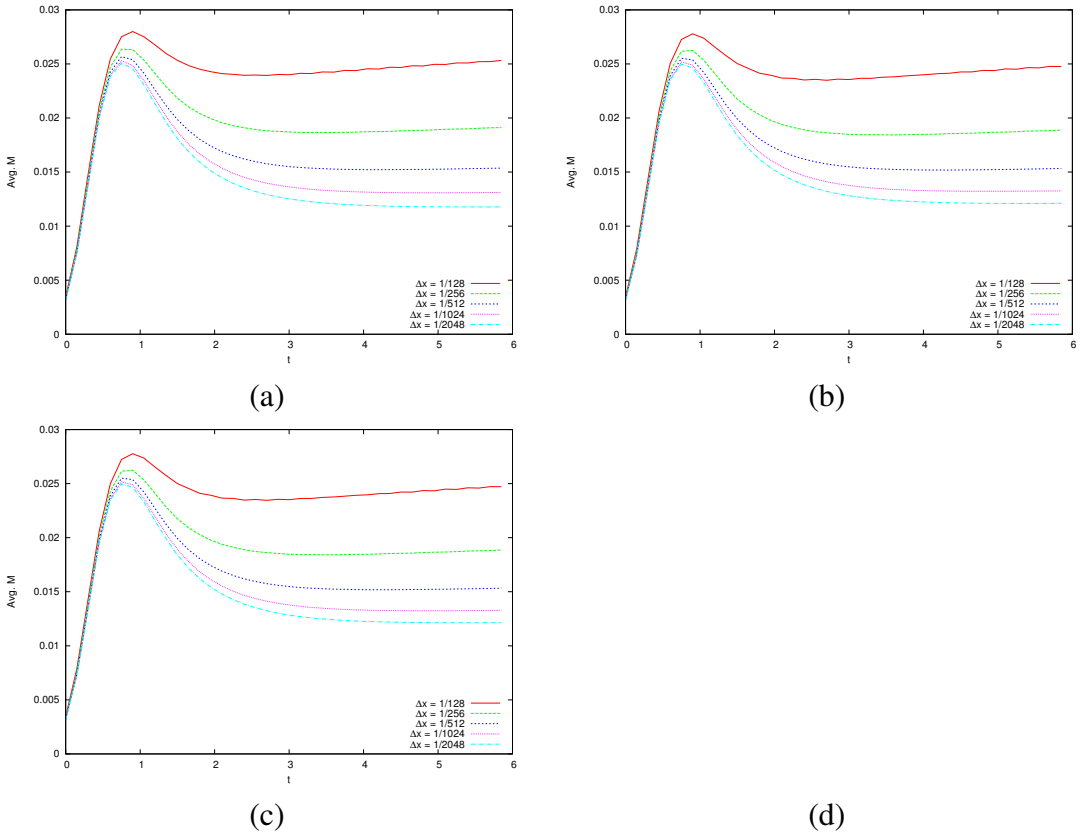


Figure 2.9: Convergence of solutions for (a) $\Delta t = 10^{-2}$, (b) $\Delta t = 10^{-3}$, (c) $\Delta t = 10^{-4}$, (d) $\Delta t = 10^{-5}$.

Figure 2.10 shows the convergence of solutions if the computation of M depends on a arbitrary function, $\hat{C}$, instead of the substrait concentration.

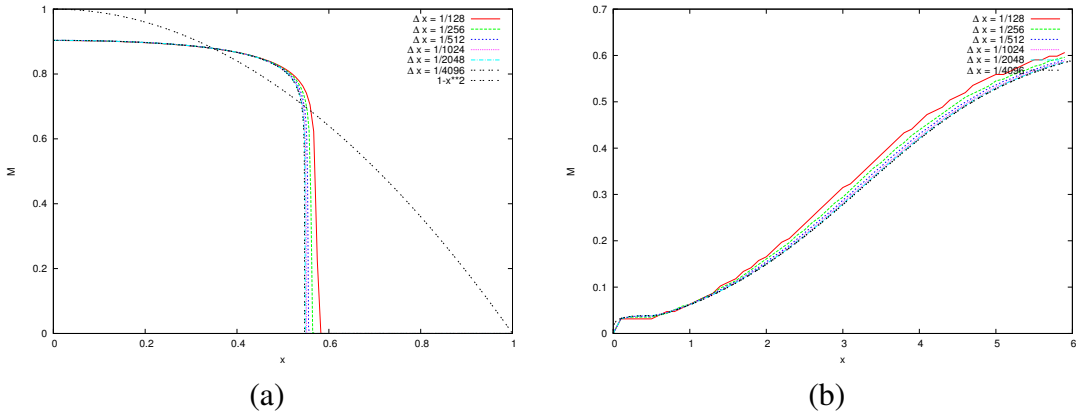


Figure 2.10: Convergence of solutions for $\hat{C}(x) = 1 - x^2$ looking at (a) solutions at $t = 5.25$ and (b) the interface x position.

Figure 2.11 show how the solutions converge for different functions $f(C, M)$. Some of which do not

depend on C or M.

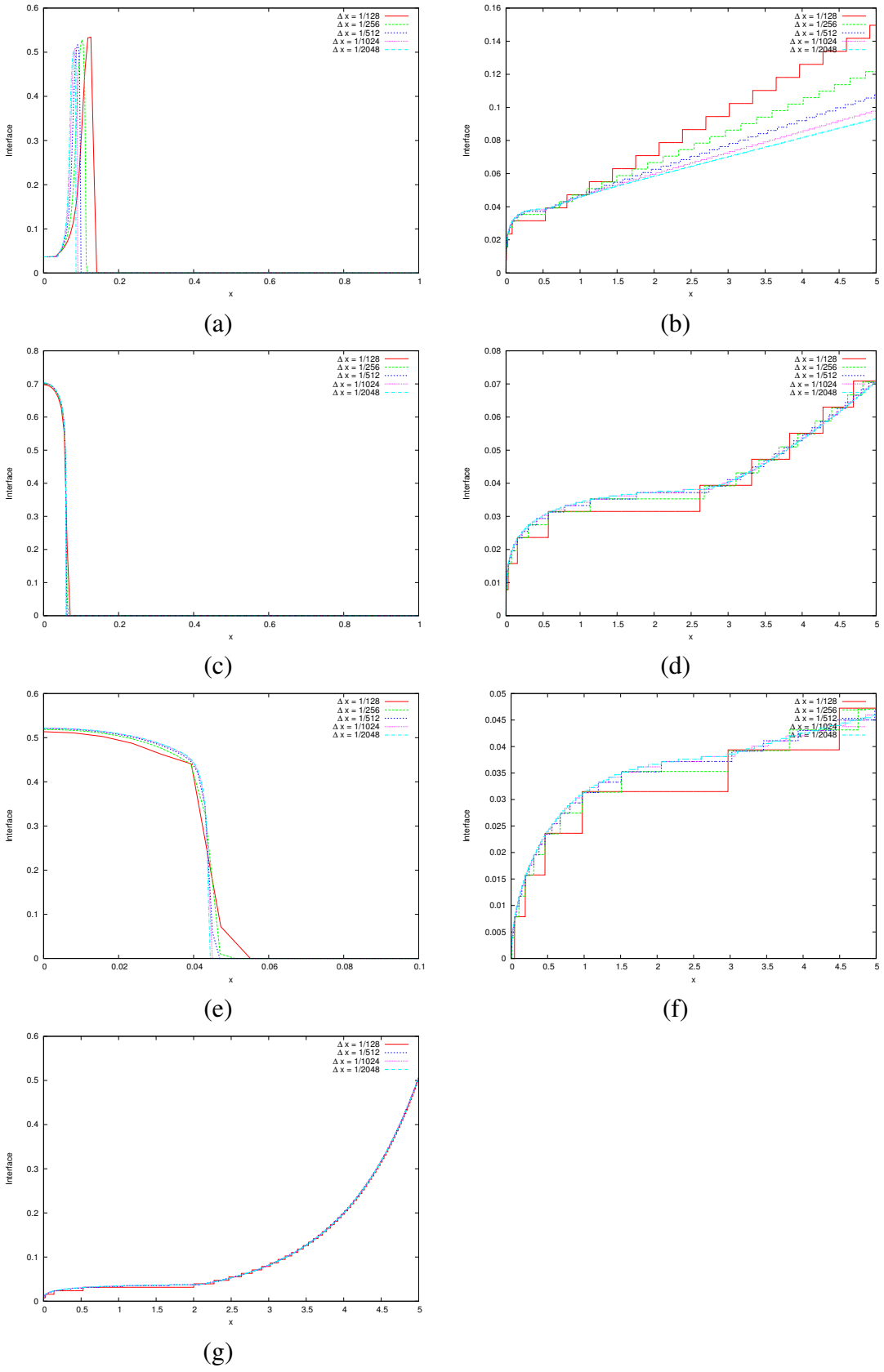


Figure 2.11: Convergence of solutions for (a-b) $f(C,M)$ =default, (c-d) $f(C,M) = (1-M)$, (e-f) $f(C,M) = (1-C)\frac{C}{M+\epsilon}$,(g) $f(C,M) = 1$. Each pair of graphs is the solution at $t = 4.25$ and the interface as a function of t.

Chapter 3

Simulation Results

3.1 Typical Simulation

Is Typical_simulation.tex needed in this section? If anything it would be the sanity check simulations

The system

$$M_t = \nabla_x \left(D(M) \nabla_x M \right) + f(C, M) M \tag{3.1}$$

$$C_t = -g(C, M) \tag{3.2}$$

where

$$D(M) = \delta \frac{M^\alpha}{(1 - M)^\beta} \tag{3.3}$$

$$f(C, M) = \frac{C}{k + C} \left(1 - \left(\frac{M M_0}{C C_0 + \epsilon} \right)^\gamma \right) \tag{3.4}$$

$$g(C, M) = \frac{\nu C}{k + C} M \tag{3.5}$$

is solved on a rectangular region with length L and width λL with the following parameter values,

$$\begin{aligned} L &= 0.01 \\ \lambda &= \frac{1}{128} & C_0 &= 30 \\ \epsilon &= 10^{-8} & M_0 &= 30 \\ \alpha &= 4 & \delta &= \frac{10^{-7}}{\mu L^2} \approx 10^-4 \\ \beta &= 4 & k &= \frac{4}{C_0} \approx 0.1333 \\ \gamma &= 0.5 & \nu &= \frac{M_0}{0.63 C_0} \approx 1.59, \\ \mu &= 6 \end{aligned} \tag{3.6}$$

and with initial conditions

$$\begin{aligned} C &= 1 \\ M &= \begin{cases} -(\frac{h}{d^4})x^4 + h & \text{if } x < 0.04 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{3.7}$$

where $h = 0.05, d = 0.04$, representing the height and depth of the inoculation site.

A series of test was done on simulation code version $0.1.1$. These were defaulted to run with $\Delta t = 10^{-3}$ and $\Delta x = \frac{1}{256}$.

The following lists the test, and observations from each.

1. Solve the system with homogenous initial conditions and see if it stays homogenous.

    - Worked!
    - The biomass and substrate concentration at $t = 8$ did not change as grid size did, which is good.
    - Biomass and substrate concentration did change for step size, but it was converging to a specific value (Table 3.1), which makes sense and suggests that default choice of Δt is good.

Δt	Biomass Density	Substrait Conc.
10^{-1}	0.743806469443	0.731575100108
10^{-2}	0.739180637812	0.736539504220
10^{-3}	0.738699851204	0.737025027162
10^{-4}	0.738651598939	0.737073472750
10^{-5}	0.738646771985	0.737078316244

Table 3.1: Values of biomass and substrate concentration at $t = 8$. A grid size of $\Delta x = 1/256$ was used.

2. Solve the system with $f = s$, s a constant, and see if total biomass follows be^{st}, where b is the initial total biomass.

   - Works with $D(M) = 0$.
   - With $D(M) = \delta$ total biomass grows a little slower then be^{st} which becomes slightly noticable at later times. The difference is less then 1%, so this looks good. See Figure 3.1(ab).
   - With $D(M) = \delta M^{\alpha}$ the total biomass matchs be^{st} until the biomass density at the innoculation point becomes greater then 1. Since physically this should never occur, this isn't a problem. See Figure 3.1(c).
   - With $D(M) = \frac{\delta M^{\alpha}}{(1-M)^{\beta}}$ the total biomass grows accuratly until it approches $M \in (0.9, 1)$ at the innoculation point. This is a problem, because at this point the density dependent diffusion should start and keep the solution growing with be^{st}. This suggests that there is something wrong with how the diffusion is being implemented, will follow up on this. See Figure 3.1(d).

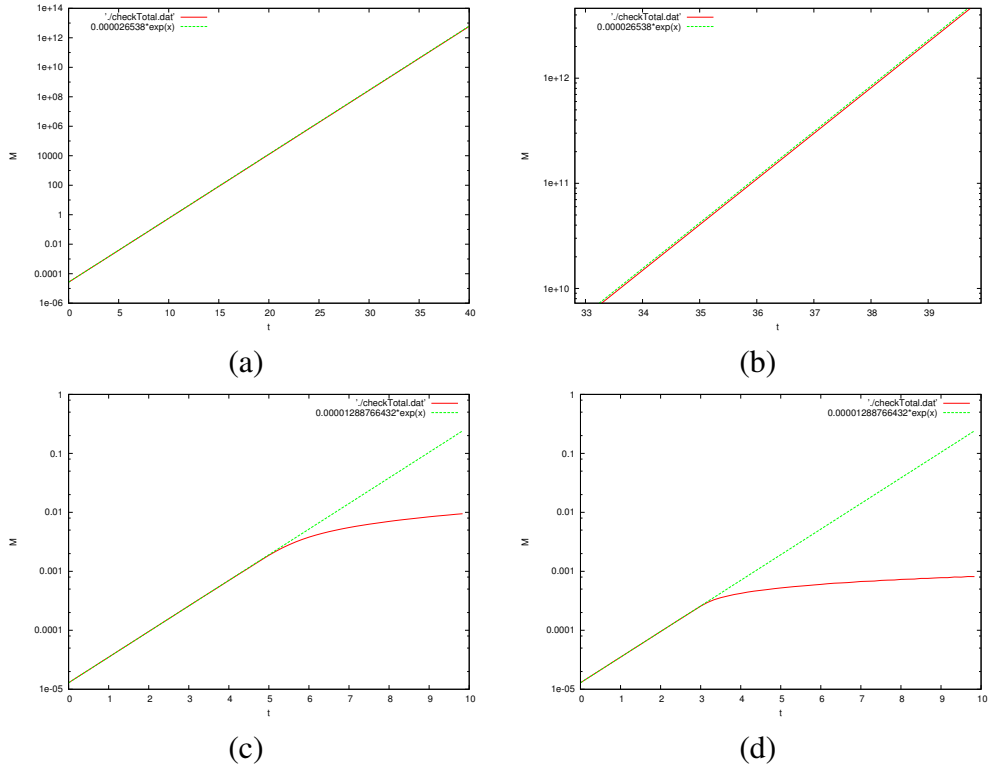


Figure 3.1: Graph of,

(a) $D(M) = \delta$,

(b) $D(M) = \delta$; zoomed in to show the slight difference in growth,

(c) $D(M) = \delta M^{\alpha}$,

(d) $D(M) = \frac{\delta M^{\alpha}}{(1-M)^{\beta}}$.

Chapter 4

Discussions

Lots of discussion on the things that were analysied in this thesis.

Chapter 5

Conclusions

There will be lots of concluding statements in here.

References

H.J. 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.

H.J. Eberl, D.F. Parker, and van Loosdrecht M.C.M. A new deterministic spatio-temporal continuum model for biofilm development. *Journal of Theoretical Medicine*, pages 161–175, 2001.