

1 Comparison of a Semi-Implicit and a Fully-Implicit Time Integration Method for a
2 Highly Degenerate Diffusion-Reaction Equation Coupled with an Ordinary
3 Differential Equation

4 by

5 Eric M. Jalbert

6 A Thesis
7 presented to
8 The University of Guelph

9 In partial fulfilment of requirements
10 for the degree of
11 Master of Science
12 in
13 Applied Mathematics

14 Guelph, Ontario, Canada

15 © E.M. Jalbert, January, 2015

¹⁶ Contents

¹⁷	1	Introductions	1
¹⁸	1.1	Background	1
¹⁹	1.2	Objectives	7
²⁰	1.3	Thesis Outline	8
²¹	2	Model Definition	10
²²	2.1	Model Description	10
²³	2.2	Nondimensionalization	14
²⁴	3	Numerical Methods	17
²⁵	3.1	Discretization	17
²⁶	3.2	Solution Technique	19
²⁷	3.3	Computational Setup	26
²⁸	3.4	Method Validation	27
²⁹	3.5	Comparison of Semi-implicit and Fully-implicit Method	34
³⁰	4	Simulation Results	36
³¹	4.1	Typical Simulation	36
³²	4.2	Travelling Wave Analysis	41
³³	4.3	Spatial Effects	51
³⁴	5	Conclusions	57
³⁵	5.1	Lessons Learned	57
³⁶	5.2	Future Work	58
³⁷		Bibliography	59
³⁸	A	Default Parameter Values	62

³⁹ **Chapter 1**

⁴⁰ **Introductions**

⁴¹ **1.1 Background**

⁴² Bacterial biofilms are microscopic deposition of organisms that embed themselves on immersed sur-
⁴³ faces whenever environmental conditions can sustain microbial growth. These bacteria, when sessile,
⁴⁴ surround themselves in a self-produced viscous layer of extracellular polymeric substances. As a
⁴⁵ result of this, the cells are extraordinarily resistant to mechanical washout or antibiotic attacks. Most
⁴⁶ bacterial populations live in communities withing the extracellular polymeric substances. They can be
⁴⁷ found in many different aspects of life in both positive ways (wastewater treatments, soil remediation,
⁴⁸ and groundwater protection) and negative ways (bacterial infections, dental plaque, biocorrosion of
⁴⁹ facilities and water pipes).

⁵⁰ The recent field of energy biotechonology has lead to researching biofilms as a potential means of
⁵¹ using plant biomass to generate sustainable fuels. *Clostridium Thermocellum* is a possible choice for
⁵² achieving large scale biomass conversion. Because of this there has been a surge of studies based
⁵³ around its behaviours and characteristics. An interesting feature of *C. Thermocellum* is that it grows
⁵⁴ as a thin cellulolytic monolayer and does not produce any extracellular polymeric matrix (Dumitrache
⁵⁵ et al., 2013a). This is a stark contrast to typical biofilms which are notorious for forming complex
⁵⁶ mushroom or pillar shaped morphologies.

57 Dumitrache (2014) have made a number of *in situ* and *in vitro* observations for *C. Thermocellum*.
58 Here they linked the cellulose consumption rate of the bacteria to the rate of CO_2 produced. The
59 experiments ran in a continuous-flow reactor that used cellulose paper sheets innoculated with *C.*
60 *Thermocellum* strains. The bacteria consumed the fibers of the cellulose sheets as they grew. By
61 tracking only the CO_2 production, they were only focused on the activity of the bacteria at a reactor-
62 scale. The smaller spatial movements of the bacteria were ignored for simplicity of the experiment.

63 Originally, biofilm models were poised as ordinary differential equations or one-dimensional partial
64 differential equations that assumed the biofilms developed as a flat layer. This simplified the calcu-
65 lation for the speed of propagation but was limited in non-spatially heterogenous biofilm morpholo-
66 gies. To this end, many models for spatially heterogenous biofilms were proposed. These included
67 stochastic individual based models, cellular automata models, and deterministic continuum models.
68 The added complexity helped in modelling more of the multi-dimensional aspects of biofilms, namely
69 the intrinsic structures that most biofilms grew. These models considered the changes the biofilms
70 made at the meso-scale instead of the reactor-scale which simpler models tended to do. The reactor-
71 scale models attributed the spatial effects by using logistic-like growth terms to limit the amount of
72 bacteria activity in the models.

73 A simple mathematical model for cellulolytic biofilm activity and growth on model cellulosic substrate
74 was proposed in Dumitrache (2014). Here the production of carbon dioxide was used as an indicator
75 of cluture metabolism. Because of this indicator, they focused on overall biofilm performance rather
76 than on detailed biofilm structure. The model formed was based on a number of observations on the
77 metabolic acitivity gathered by online carbon dioxide measurements. For the model formation, atten-
78 tion was also put on high-resolution imaging of different stages of biofilm development (Dumitrache
79 et al., 2013a) and to the physiological behaviour of substrate modification (Dumitrache et al., 2013b).

80 The conceptual model of cellulolytic biofilm growth that was followed for their model is shown in
81 Figure 1.1. This model is based on the relation between *C. Thermocellum* growth and Whatman cel-
82 lulose sheet consumption. The relation is the idea that as the *C. Thermocellum* grows more cellulose

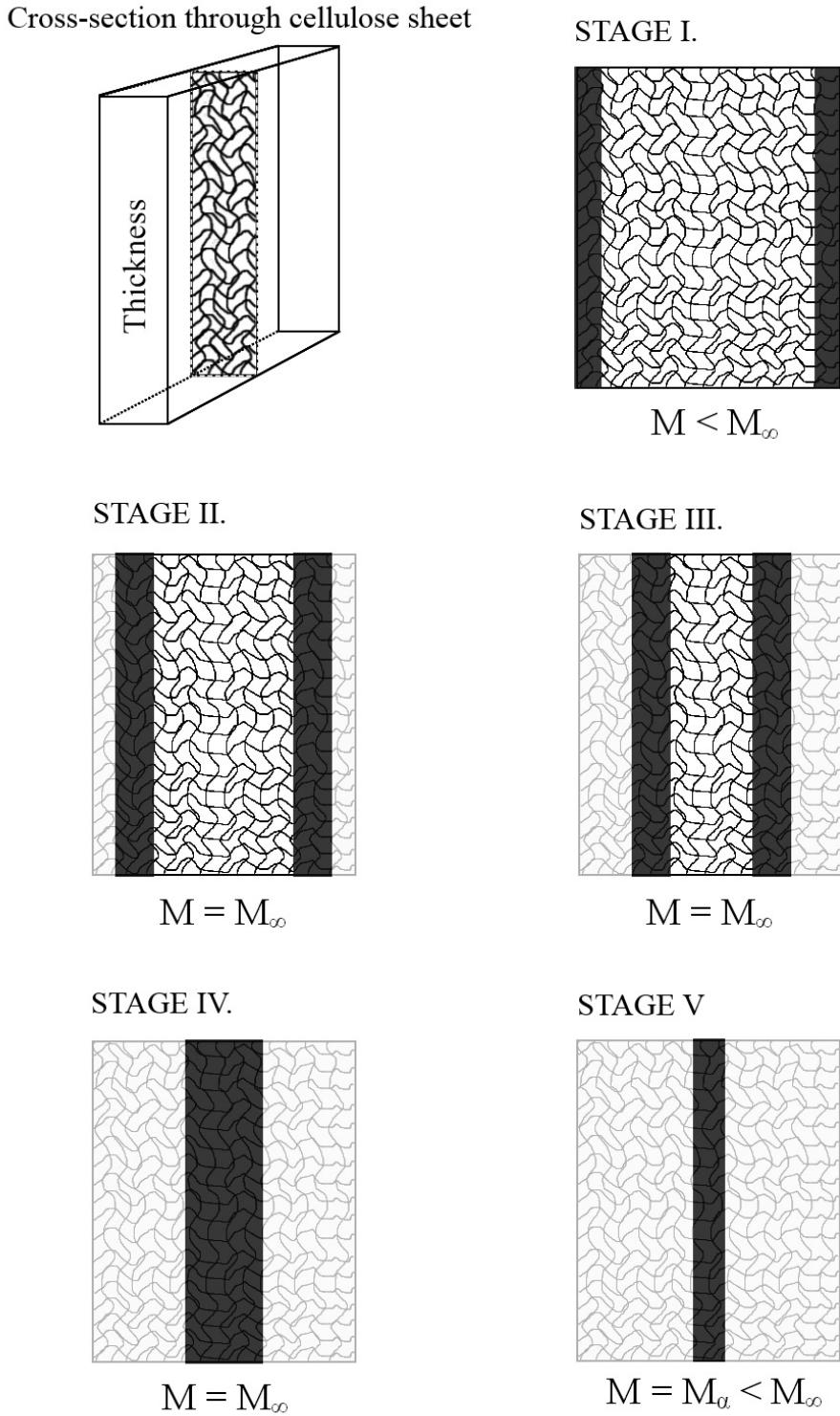


Figure 1.1: Conceptual model of cellulolytic biofilm growth and consumption of cellulose sheets. Attachment and growth occurs on both sides of the sheet, individual monolayers form on each fiber and result in a band of active biofilm (i.e., the effective sessile biomass) M (dark band). The ideal carrying capacity, M_{∞} , and the actual carrying capacity, M_{α} , are explained in Dumitrache (2014). Consumed substrate is represented by the light gray areas. Figure originally from Dumitrache et al. (2015).

sheet is consumed which limits the area of substratum available for bacteria attachment. Here they express the growth of the biomass in terms of an *ideal* and an *actual* carrying capacity. The ideal carrying capacity, M_∞ , refers to the maximum amount of biomass that can be supported when the only limitations are from a deficiency of local space. This value depends on the properties of the substratum and is assumed to be constant since it is independent of the substrate concentration. The actual carrying capacity, M_α , references the amount of biomass that can be support when the local concentration of substrate mass is limited. This value is not constant since it is a function of the current substrate concentration. In essence, Dumitrache (2014) use the carrying capacity as a means to account for the limitations due to spatial effects.

The model developed by Dumitrache et al. (2015) followed the conceptual five different stages of growth of *C. Thermocellum*. These five stages were:

- Stage I: Independent colonies of cells grow on the matrix of fibers in the substrate. This occurs initially for all the isolated regions of the biofilm.
- Stage II: The biofilm grows inwards. The superficial fibres are consumed and newly unsupported biofilms are released from the cellulos sheet into the aqueous stream.
- Stage III: The active biofilm band stabalizes somewhere around the point where superficial fiber deconstruction rate is the equivalent to the inwards penetration rate of the biofilm.
- Stage IV: The progression of the biofilm band continues until the remaining amount of useable substrate becomes limited.
- Stage V: The remaining cellulose gets consumed without new biofilm being produced. Instead a new generation of non-adherent cells is formed locally.

This formulated a system of ordinary differential equation because of the non-diffusivity of the substrate. These ordinary differential equations resembled traditional growth models in batch cultures more then the typically complex biofilm model seen in studies (Wanner et al., 2005).

107 The model developed in Wang et al. (2011) focused more on detailing the spatio-temporal devel-
108 opment of a single *C. Thermocellum* biofilm colony. This was completed by use of a nine-neighbor
109 square cellular automata on a 30×15 grid with a single grid cell as an inoculation point. The model
110 results matched the experimental results they gathered. This was the first model to consider the de-
111 velopment of *C. Thermocellum* at a small scale. Using cellular automata with such a coarse grid gave
112 them a discrete representation of the system they modelled. In this thesis, one purpose is to extend
113 this concept to a continuous model similar to Eberl and Demaret (2007) but instead using assumptions
114 and growth function that match the behaviour of *C. Thermocellum*.

115 The problem with this is that with *C. Thermocellum* the substratum used for attachment is consumed
116 with biofilm growth. This, along with the stationary nature of the cellulose sheets, makes this prob-
117 lem significantly different from other biofilm models. Other biofilm models are based on the aqueous,
118 free-floating environment where biofilms develop. At the meso-scale, our *C. Thermocellum* system
119 must model the development along the non-diffusing individual fibers of the cellulose sheet struc-
120 ture. Thus, these two categories of biofilm models differ mainly in their consideration of substrate
121 diffusion, with *C. Thermocellum* there is none.

122 Our problem originates from the ordinary differential equation model from Dumitrache et al. (2015).
123 Here we include the double-degenerate parabolic model of biofilm formation from Eberl et al. (2001)
124 for the spatial consideration. This results in a density-dependent diffusion model with reaction terms
125 that match with the behaviour of *C. Thermocellum*. The spatial operator for biofilm spreading shows
126 two non-standard diffusion effects:

- 127 ● A power law degeneracy similar to the porous medium equation for local biomass at the inter-
128 face of the biofilm Gurtin (1977)

- 129 ● A singularity in the diffusion coefficient when the biofilm approaches maximum biomass den-
130 sity.

131 Both of these effects together lead to the development of sharp and steep interfaces in the model

132 solutions that mark the separation of the actual biofilm from its liquid environment. Such propagating
133 interface problems in partial differential equations often are difficult to treat numerically.

134 From a mathematical point of view, Efendiev et al. (2002) showed that models which focus on the
135 growth dynamics of biological film are mostly in the form of systems of partial differential equations.
136 These models take into account some of the significant features of biofilm growth observed throughout
137 the practice. These features are:

- 138 a) the presence of a sharp front of biomass at the solid to fluid transition region,
- 139 b) the existence of a threshold of biomass density,
- 140 c) the spreading of biomass is only notable when local densities approach the threshold of sustain-
141 ability,
- 142 d) the use of reaction kinetics mechanisms to model the production of biomass,
- 143 e) the compatibility of biomass spreading with nutrient transfer and consumption mechanism models.

144 There exist mathematical background for these systems of equation that prove existence and unique-
145 ness of positive and bounded solutions. However, the complexity of these nonlinear partial differ-
146 ential equations have made practically impossible to provide analytical expressions of the solutions
147 given biologically meaningful initial conditions. This forces numerical computational approaches to
148 be taken for simulation the growth of these microbial colonies. Some techniques have been used for
149 these problems in stochastic computational modelling McCollum et al. (2006). The finite-difference
150 approach has been proposed in Eberl et al. (2001) for deterministic models. Differential-discrete cel-
151 lular automata have been successfully used for the growth of gel beads in Picioreanu et al. (1998) The
152 finite-element used in Duddu et al. (2008) and the finite-volume method used in Gallo and Manzini
153 (2003) were both able to handle the complexity of the problem. The finite-different method was
154 used as an explicit numerical method in Macías-Díaz et al. (2013) for a faster numerical computation
155 compared to Eberl et al. (2001).

156 In this thesis, the finite-difference method is treated as a semi-implicit numerical method similar to
157 Eberl et al. (2001). However a new extension is made on this where in a single timestep multiple
158 solutions are found using a fixed-point iteration. The use of this is that the terms of the system are
159 split into forward difference terms and backwards difference terms. The forward difference terms are
160 those that make finding an analytic solution difficult. For each iteration of the fixed-point iteration,
161 the forward difference terms are updated to a new approximation of their *a priori* value. This turns
162 the semi-implicit method into a fully-implicit method since the next timestep is reached once all the
163 forward difference terms become sufficiently close to their true value for the next timestep. The
164 validity of this method is unknown and put under scrutiny during this thesis.

165 **1.2 Objectives**

166 There are three main objectives of this thesis:

- 167 1. The numerical development, implementation and validation of the proposed fully-implicit method.
168 This includes both the theoretical and computational verification of its solutions. Theoretically
169 the method must be shown to converge to a single true solution. Computationally there are a
170 number of issues that could arise from the implementation: sinks or sources of biomass could be
171 introduced, certain characteristics could be lost, and mass may no longer be conserved. These
172 each were investigated for their existance in our implementation. Also, a spatial discretization
173 convergence analysis was conducted to verify that the given solutions agree with selection of
174 grid size.
- 175 2. The comparison of the fully-implicit method and the semi-implicit method. This is for taking
176 inventory of the changes that the newly developed fully-implicit method has against the semi-
177 implicit method for which it is based upon. Multiple aspects of the solutions are compared to
178 verify the added computational effort needed. A recommendation on the use for this method is
179 established only after the comparison is made.
- 180 3. Numerous simulations are conducted to further the understanding of the mechanisms of *C*.

181 *Thermocellum*. The normal behaviour of the system is observed from running a typical simula-
182 tion. Some interesting observations are made from further simulations: travelling wave solution
183 existance and different spatial effect from the choice of intial condition.

184 1.3 Thesis Outline

- 185 • Background (1.1): Describes the placement of this problem in the existing literature.
- 186 • Objectives (1.2): Describes the three main objectives of this thesis.
- 187 • Outline (1.3): Outlines the sequencial format of the thesis.
- 188 • Model Description (2.1): Formulates the mathematical model that is used for the rest of the
189 study.
- 190 • Nondimensionalization (2.2): The system described in the previous section is reduced to unit-
191 less parameters and variables for simplicity.
- 192 • Discretization (3.1): Spatial and temporal discritizations are applied to the system.
- 193 • Solving Technique (3.2): Using a fixed-point iteration, solutions for the system are found by use
194 of finite difference method with the Bi-Conjugate Gradiant Method and the quadratic equation.
- 195 • Computational Setup (3.3): The work stations and compiler settings are recorded.
- 196 • Method Validation (3.4): Multiple test are run to check if the handling of the spatial discritiza-
197 tion generated any sources or sinks in the solution. A measure to compare two the solution of
198 two different simulations is esstablished so that a grid convergence analysis can be done.
- 199 • Comparison of Semi-Implicit and Fully-Implicit Methods (3.5): Using the previously estab-
200 lished measure, the semi-implicit method and fully-implicit method are compared against vary-
201 ing quantities and at different tolerances.
- 202 • Typical Simulation (4.1): A generic simulation is run to reveal the general behaviour of the
203 system.

- 204 ● Travelling Wave Analysis (4.2): A travelling wave solution is suggested to exist, this is tested
205 against different simulations to reinforce the idea.
- 206 ● Spatial Effects (4.3): The effect of ignoring the spatial diffusion is investigated by comparing
207 two vastly different simulations; one with evenly distributed initial condition, the other with
208 clumped initial condition.
- 209 ● Lessons Learned (5.1): The general findings of each section are summarized.
- 210 ● Future Work (5.2): Possible changes and extensions are discussed.

211 **Chapter 2**

212 **Model Definition**

213 **2.1 Model Description**

214 The model used for simulations is based on the deterministic biofilm model developed in Eberl et al.
215 (2001), which was designed for modelling the development of spatially heterogenous biofilm struc-
216 tures. Since *C.Thermocellum* grows as a monolayered biofilm and consumes a solid carbon fiberous
217 substrate, there are mechanical differences between the two systems. Our model is based on the
218 following assumptions:

- 219 1. The growth of sessile biomass is limited locally by the availability of nutrients and by the
220 availability of colonizable space.
- 221 2. The number of cells per unit area of substratum is limited to a finite value because *C.Thermocellum*
222 forms only a thin monolayer.
- 223 3. Biomass does not spread until its density approaches the physical limit. Near the physical limit
224 it expands spatially into neighbouring regions. Because of this, the physical limit of biomass
225 density is never attained.
- 226 4. The carbon fiberous substrate consumed as nutrition for biomass growth is the substratum to
227 which the biofilm attaches. Carbon is bound in the fibers of the substratum and does not diffuse.

228 The propagation of biofilm is based on the lack of available substrate locally, not the physical
 229 degradation of the substratum.

230 5. Cell death and cell loss into the aqueous environment is assumed to be proportional to cell
 231 density.

232 6. Biomass growth is proportional to substrate consumption.

233 Assumption 1, 2, 3, 5, and 6 are similar to those made in Eberl et al. (2001). The main difference
 234 here is 4; our substrate is sessile. With a sessile substrate, there is no diffusion for the substrate
 235 concentration. Another difference is that *C. Thermocellum* does not grow from the substratum into
 236 the aqueous phase. Instead our biofilm grows across the substratum making this a two dimensional
 237 setting.

238 The model is formulated in a spatial domain Ω . The independent variables $t > 0$ denote time and
 239 $x \in \Omega$ denotes the location within the physical domain. The dependent variables are the local
 240 fraction of the surface occupied by biomass $M(t, x)$ and the substrate density $C(t, x)$. The net growth
 241 rate of biomass, in dependence of available substrate we denote by $f(C)$, the substrate consumption
 242 rate by $g(C)$. The diffusion coefficient that describes spatial expansion of biomass is given by the
 243 function $d(M)$.

244 From the above assumptions, a PDE-ODE-coupled system that models *C. Thermocellum* growth on
 245 carbonous fibres can be formulated as,

$$M_t = \nabla_x (d(M) \nabla_x M) + f(C)M \quad (2.1)$$

$$C_t = -g(C)M \quad (2.2)$$

246 where

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

248

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

249

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

250

251 with all parameters non-negative. Here we have a pair of equations, (2.1) and (2.2), that represent
 252 the biomass density and substrate concentration respectively. This is a model for the spatial spreading
 253 of biomass. For $0 < M \ll 1$ the spreading effect is negligible but when $0 \ll M \approx 1$ we have
 254 considerable spreading. The work done by Khassehkhan et al. (2009) shows this to be an ideal choice
 255 for modelling the spreading of biofilms. By assumption 1, the only factors effecting the biomass
 256 density is growth from nutrient conversion and diffusion from local spatially-full colonized space.
 257 For the density-dependent diffusion equation (2.3), δ is the diffusion coefficient, which controls the
 258 magnitude of this term, and the parameters α and β are selected to control the strength of the diffusion.
 259 This agrees with assumption 3 since (2.3) has a near-zero value until when $M \rightarrow 1$, which leads to
 260 $d(M) \rightarrow \infty$ as seen in Figure 2.1. The production rate is the difference between simple Monod
 261 kinetic growth term, with growth rate u , and a constant death rate term, n , to agree with assumption
 262 1 and 5. Monod kinetic growth was selected, with half-saturation carbon concentration k , since it
 263 matches the growth of bacteria when limited by available nutrients.

264 Equation (2.2) describes the consumption of carbon substrate due to biomass growth. Parameter y
 265 is the maximum consumption rate, measured in mass carbon per unit time. Substrate consumption
 266 is proportional to the local biomass density M . Parameter k , same as in the growth term for (2.1)
 267 is again the half-saturation carbon concentration. Here assumption 4 and 6 are satisfied since there
 268 exist no diffusion term for the substrate and its growth is a scalar multiple of the biomass growth rate.

269 The dimensions of the parameters and variables are in Table 2.1. Note that since we have a two
 270 dimensional problem, due to the lack of complex biofilm structures from *C. Thermocellum* growth,
 271 the spatial considerations are all strictly for area and not volume, as is typically done for biofilm
 272 modelling.

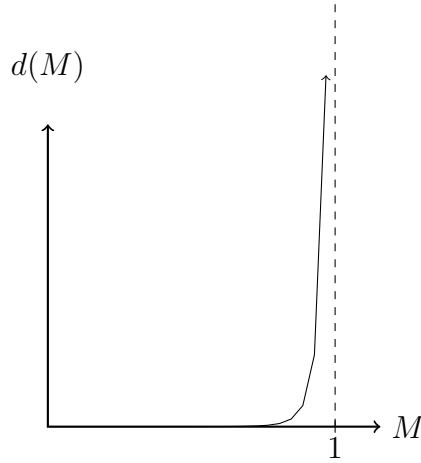


Figure 2.1: A graph of $d(M) = d \frac{M^\alpha}{(1-M)^\beta}$ showing the way diffusion increases asymptotically as $M \rightarrow 1$.

Description	Symbol	Dimensions
Spatial region	Ω	NA
Time	t	[days]
Location in Ω	x	[meters]
Biomass fraction	M	[—]
Substrate concentration	C	[$\frac{\text{grams}}{\text{meters}^2}$]
Diffusion coefficient	d	[$\frac{\text{meters}}{\text{days}}$]
Density-dependent exponent	α	[—]
Density-dependent exponent	β	[—]
Growth rate	u	[days $^{-1}$]
Half-saturation carbon concentration	k	[$\frac{\text{grams}}{\text{meters}^2}$]
Maximum consumption rate	y	[$\frac{\text{grams}_{\text{carbon}}}{\text{days}}$]
Death constant	n	[$\frac{\text{grams}}{\text{meters}^2 \cdot \text{days}}$]

Table 2.1: List of parameters and their dimensions

273 The model (2.1), (2.2) is completed by boundary conditions for the biomass density, M , and ini-
 274 tial conditions for both M and substrate concentration C . For M we pose homogenous Neumann
 275 boundary conditions such that,

276
$$\partial_n M = 0, \quad x \in \partial\Omega. \quad (2.6)$$

277 The initial conditions for the biomass density is,

278
$$M(0, x) = M_0(x), \quad x \in \Omega, \quad (2.7)$$

279 where $0 \leq M_0(x) < 1$ and $M_0(x)$ non-zero in specific pockets on the substratum. These are specified
 280 below for each individual simulation experiments. The initial conditions for the substrate concentra-
 281 tion is,

282
$$C(0, x) = C_\infty, \quad x \in \Omega, \tag{2.8}$$

283 where C_∞ describes the initial carbon density in the substratum.

284 There has been shown to exist a finite speed of interface propagation for the solutions of these kinds
 285 of degenerate problems, where $d(0) = 0$ and $\alpha > 1$ from (2.3) (Jalbert and Eberl, 2014). These
 286 problems have a blow up in the biomass gradient at the interface because of the degeneracy that exists
 287 there. For this system, we have $M < 1$ always since the diffusion when $M \approx 1$ is great enough to
 288 always ensure this.

289 **2.2 Nondimensionalization**

290 To help facilitate the analysis of this system, the full removal of all physical units is preferred and
 291 so we nondimensionalize the parameters. Here the parameters used are: the biomass growth rate, u ;
 292 the length of the region, L ; and the maximum density for biomass and substrate, M_∞ and C_∞ . The
 293 biomass density fraction represents the current density of biomass divided by the maximum biomass
 294 density, M_∞ . From using the following parameter changes, the system can be made unitless.

$$\chi = \frac{x}{L} \implies Ld\chi = dx \quad (2.9)$$

$$\tau = ut \implies \frac{1}{u}d\tau = dt \quad (2.10)$$

$$\mathcal{C} = \frac{C}{C_\infty} \quad (2.11)$$

$$\delta = \frac{1}{uL^2}d \quad (2.12)$$

$$\kappa = \frac{k}{C_\infty} \quad (2.13)$$

$$\nu = \frac{n}{uC_\infty} \quad (2.14)$$

$$\gamma = \frac{M_\infty}{C_\infty}y \quad (2.15)$$

²⁹⁵ Using these, (2.1) and (2.2) can be simplified and nondimensionalized into,

$$M_\tau = \nabla_\chi (D(M)\nabla_\chi M) + F(\mathcal{C})M \quad (2.16)$$

$$\mathcal{C}_\tau = -G(\mathcal{C})M, \quad (2.17)$$

²⁹⁶ where,

$$\begin{aligned} D(M) &= \delta \frac{M^\alpha}{(1-M)^\beta} \\ F(\mathcal{C}) &= \frac{\mathcal{C}}{\kappa + \mathcal{C}} - \nu \\ G(\mathcal{C}) &= \gamma \frac{\mathcal{C}}{\kappa + \mathcal{C}}. \end{aligned} \quad (2.18)$$

²⁹⁸ with only $\delta, \kappa, \nu, \gamma$ as model parameters. For convenience, we henceforth use

$$C := \mathcal{C}, \quad x := \chi, \quad t := \tau. \quad (2.19)$$

300 Each of the dimensionless parameters in (2.18) have a biological representation based on the trans-
301 formations done. The parameter δ is the dimensionless biomass motility coefficient. It affects the
302 change in biomass from adjacent biomass sources, a greater δ results in faster biofilm expansion.
303 The parameter κ is the half-saturation point, it is exactly the value for which substrate concentration
304 results in 0.5-optimum growth rate. Parameter ν is the decay and loss rate for biomass. These can
305 be from starvation in cases where substrates are depleted or from loss into the aqueous environment.
306 Lastly, γ is the dimensionless maximum substrate consumption rate. It signifies the ratio of substrate
307 consumed to biomass growth. Here, a larger γ value results in more substrate being consumed to
308 produce the same amount of biomass.

309 With (2.16) being reduced to these parameters the numerical analysis become more simplified while
310 still retaining the same significance in results.

311 **Chapter 3**

312 **Numerical Methods**

313 **3.1 Discretization**

314 In order to approximate the solution for (2.16) spatial and temporal discretizations must be made.

315 First the equations are discretized in time,

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

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

319 Here, (3.1) follows the ideas of the Backwards Euler Method; (3.2) follows Trapezoidal Rule (Burden
320 and Faires, 2010). The index variable k has been introduced in (3.1) - (3.2) such that $M^k(x) \approx$
321 $M(t^k, x)$, allowing an approximation at a certain time, t^k , to be used; this changes the spatial-temporal
322 continuum model into a spatial continuum model with discrete temporal timesteps.

323 For this system, we let the nondimensional spatial region of consideration, $\Omega = [0, 1] \times [0, 1]$, be
324 square. Now, only (3.1) requires spatial considerations since the substrate does not diffuse across the
325 region. The spatial discretization will be through the Finite Difference Method as described in Saad
326 (2003). Here, a uniform $n \times m$ grid is used to discretize Ω . Since all the calculations will be done on
327 the grid intersections the discretization will be grid-point based. This means that a $n \times m$ grid implies

328 there are $(n - 1) \times (m - 1)$ grid boxes. The distance between grid points is the same in both x_1 and
 329 x_2 dimensions; we have $\Delta x_1 = \Delta x_2 = \Delta x$. Since we work on a nondimensionalized domain, and
 330 we know the number of grid boxes in our region, we have that $\Delta x = \frac{1}{n-1}$. A five-point stencil is
 331 used to approximate the solution of (3.1) at each grid point. This spatial discretization allows the use
 332 of i and j to index across the region such that $x_{1_i} = i * \Delta x$ for $i \in \{0, 1, \dots, n - 1\}$ and $x_{2_j} = j * \Delta x$
 333 for $j \in \{0, 1, \dots, m - 1\}$. To index the grid point, i and j are used such that $M_{i,j}^k \approx M(t^k, x_{1_i}, x_{2_j})$.
 334 To account for the dependency on neighbouring grid points, we introduce σ as the index pair from the
 335 set

$$336 \quad \mathcal{N}_{ij} = \{n_{ij}, e_{ij}, s_{ij}, w_{ij}\}. \quad (3.3)$$

337 where,

$$338 \quad \begin{aligned} n_{ij} &= \begin{cases} (i, j + 1) & \text{if } j < m \\ (i, j - 1) & \text{if } j = m \end{cases} & e_{ij} &= \begin{cases} (i + 1, j) & \text{if } i < n \\ (i - 1, j) & \text{if } i = n \end{cases} \\ s_{ij} &= \begin{cases} (i, j - 1) & \text{if } j > 0 \\ (i, j + 1) & \text{if } j = 0 \end{cases} & w_{ij} &= \begin{cases} (i - 1, j) & \text{if } i > 0 \\ (i + 1, j) & \text{if } i = 0 \end{cases}. \end{aligned} \quad (3.4)$$

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

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

$$341 \quad \frac{M_{i,j}^{k+1} - M_{i,j}^k}{\Delta t} = \frac{1}{\Delta x^2} \sum_{\sigma \in \mathcal{N}_{ij}} \left(\frac{D(M_\sigma^{k+1}) + D(M_{i,j}^{k+1})}{2} \right) \cdot (M_\sigma^{k+1} - M_{i,j}^{k+1}) + F(C_{i,j}^{k+1}) M_{i,j}^{k+1} \quad (3.5)$$

342 For completeness, we spatially discretize (3.2) as

$$343 \quad \frac{C_{i,j}^{k+1} - C_{i,j}^k}{\Delta t} = \frac{1}{2} (G(C_{i,j}^{k+1}) M_{i,j}^{k+1} + G(C_{i,j}^k) M_{i,j}^k). \quad (3.6)$$

344 Notice that for (3.5), the arithmetic mean of the diffusion function, D , is taken because of the steep
 345 gradient at the interface. The alternative would be to use $D(M_{i+\frac{s}{2}, j+\frac{r}{2}}^{k+1})$, however in some cases we
 346 have $M_{i+\frac{s}{2}, j+\frac{r}{2}}^{k+1} = 0$ which would result in $D(0) = 0$ and thus nullify the effect of the spatial diffusion.

³⁴⁷ Taking the arithmetic mean eliminates this result because the average value of D would not be zero
³⁴⁸ at the interface.

³⁴⁹ To simplify the spatial indexing, the matrix system is converted into a vector by use of a bijective
³⁵⁰ mapping defined as:

$$\begin{aligned} \pi : \quad \{0, \dots, n\} \times \{0, \dots, m\} &\rightarrow \{0, \dots, nm\} \\ \text{351} \quad (i, j) &\rightarrow \pi(i, j) \end{aligned} \quad (3.7)$$

³⁵² Now, a single index can be used to iterate over the vector, l . This gives the system,

$$\begin{aligned} \text{353} \quad \frac{M_l^{k+1} - M_l^k}{\Delta t} &= \frac{1}{\Delta x^2} \sum_{\sigma \in \mathcal{N}_{ij}} \left(\frac{D(M_\sigma^{k+1}) + D(M_l^{k+1})}{2} \right) \cdot (M_\sigma^{k+1} - M_l^{k+1}) + F(C_l^{k+1})M_l^{k+1} \end{aligned} \quad (3.8)$$

$$\begin{aligned} \text{354} \quad \frac{C_l^{k+1} - C_l^k}{\Delta t} &= \frac{1}{2}(G(C_l^{k+1})M_l^{k+1} + G(C_l^k)M_l^k). \end{aligned} \quad (3.9)$$

356 3.2 Solution Technique

³⁵⁷ Assuming the values for C and M at time level k are known, (3.8) and (3.9) are a coupled system
³⁵⁸ of $2nm$ highly nonlinear equation for $2nm$ unknown M_l^{k+1}, C_l^{k+1} . To solve this coupled system, we
³⁵⁹ define a fixed point iteration:

$$\begin{aligned} \text{360} \quad M_l^{(p)} &:= M_l^k, \quad C_l^{(p)} := C_l^k, \end{aligned} \quad (3.10)$$

³⁶¹ which we apply to (3.8) and (3.9). In a single time step, the solutions for M and C can be solved
³⁶² using the previous time step solution in the follow manner:

$$\begin{aligned} \frac{M_l^{(p+1)} - M_l^k}{\Delta t} &= \frac{1}{\Delta x^2} \sum_{\sigma \in \mathcal{N}_{ij}} \left(\frac{D(M_\sigma^{(p)}) + D(M_l^{(p)})}{2} \right) \cdot (M_\sigma^{(p+1)} - M_l^{(p+1)}) + F(C_l^{(p)})M_l^{(p+1)} \end{aligned} \quad (3.11)$$

$$\begin{aligned} \text{363} \quad \frac{C_l^{(p+1)} - C_l^k}{\Delta t} &= \frac{1}{2}(G(C_l^{(p+1)})M_l^{(p+1)} + G(C_l^k)M_l^k) \end{aligned} \quad (3.12)$$

366 where $(p) \in (0, 1, 2, \dots)$ The fixed-point iteration is stopped when convergence is achieved. This is
 367 when the difference between consecutive iterations is below a selected tolerance, i.e.

$$368 \quad \sum_l^n \left(\left| M_l^{(p+1)} - M_l^{(p)} \right| + \left| C_l^{(p+1)} - C_l^{(p)} \right| \right) < tol. \quad (3.13)$$

369 At the end of the fixed-point iteration, the number of iterations is recorded as P , and we define,

$$370 \quad M_l^{k+1} := M_l^{(P)}, \quad C_l^{k+1} := C_l^{(P)}. \quad (3.14)$$

371 In this fixed point format, given by (3.11) - (3.12), the equations can be rearranged and solved by
 372 conventional methods.

373 In each iteration step, (3.11) is a simultaneous linear system for the nm unknown $M_l^{(p+1)}$. From this
 374 a linear system of equations can be created following Saad (2003). For each grid point, l a linear
 375 system is defined as:

$$376 \quad \begin{aligned} \frac{M_l^k}{\Delta t} &= \sum_{\sigma \in \mathcal{N}_{ij}} \left(\frac{D(M_\sigma^{(p+1)}) + D(M_l^{(p+1)})}{2\Delta x^2} \cdot M_\sigma^{(p+1)} \right) \\ &+ \sum_{\sigma \in \mathcal{N}_{ij}} \left(\left(\frac{D(M_\sigma^{(p+1)}) + D(M_l^{(p+1)})}{2\Delta x^2} \right) - F(C_l^{(p)}) + \frac{1}{\Delta t} \right) M_l^{(p+1)}. \end{aligned} \quad (3.15)$$

³⁷⁷ From (3.15), a five-diagonal matrix can be created defined as,

$$A = \begin{pmatrix} A_1 & a_2 & & & a_m \\ a_1 & A_2 & a_3 & & a_{m+1} \\ \ddots & \ddots & \ddots & & \ddots \\ a_1 & & a_{n-1} & A_n & a_{n+1} & & a_{n+m} \\ \ddots & & \ddots & \ddots & \ddots & & \ddots \\ a_{(n-1)m-m} & & a_{(n-1)m-1} & A_{(n-1)m} & a_{(n-1)m+1} & & a_{(n-1)m+m} \\ \ddots & & & \ddots & \ddots & & \ddots \\ & & a_{n(m-1)-1} & & a_{nm-2} & A_{nm-1} & a_{nm} \\ & & a_{n(m-1)} & & a_{nm-1} & A_{nm} & \end{pmatrix} \quad (3.16)$$

³⁷⁸

³⁷⁹ where each A_l is the diagonal coefficient and a_l is the off-diagonal coefficient based on (3.15).

³⁸⁰ **Proposition 3.2.1.** *The matrix A is positive definite and symmetric when $\Delta t < \left(F(C_l^{(p)})\right)^{-1}$.*

³⁸¹ *Proof.* Matrix A is positive definite if all the eigenvalues are positive. Using the Circle theorem
³⁸² described by Varga (2004), the eigenvalues can be shown to be positive if, independently on all rows,
³⁸³ the sum of the off-diagonals values is less than the diagonal value. This can be verified. From (3.15)
³⁸⁴ it can be said that,

$$\sum_{\sigma \in \mathcal{N}_l} \left(\frac{D(M_\sigma^{(p)})}{\Delta x^2} \right) < \left| \left(\sum_{\sigma \in \mathcal{N}_l} \left(\frac{D(M_\sigma^{(p)})}{\Delta x^2} \right) - F(C_l^{(p)}) + \frac{1}{\Delta t} \right) \right|. \quad (3.17)$$

³⁸⁶ This simplifies to,

$$\frac{1}{\Delta t} < F(C_l^{(p)}) \quad (3.18)$$

³⁸⁸ The symmetry of A can be trivially shown if one considers the formation of the diagonals. On a single
³⁸⁹ row, each element corresponds to the adjacent grid points of grid l . As the grid ordering counts along,
³⁹⁰ the elements that are equidistant from the diagonal are actually reference to the same grid point.
³⁹¹ Therefore we have symmetry. \square

392 It is important to remark that this condition for which A is positive definite and symmetric is practi-
 393 cally not a severe constraint. The condition, $\frac{1}{F(C)} < \Delta t$, relates the growth of the biomass to the size
 394 of timestep selected. In order to resolve any biomass growth, Δt must obviously be chosen smaller
 395 then the characteristic time scale of growth, $\frac{1}{F(C)}$.

396 Given that A is positive definite and symmetric, the conjugate gradiant method can be used to compute
 397 the solution.

398 **Proposition 3.2.2.** *The matrix A is diagonally dominate when $\Delta t < \left(F(C_l^{(p)})\right)^{-1}$.*

399 *Proof.* This is trivially shown to be true when one considers (3.17). It was shown that this simplifies
 400 to

$$401 \quad \frac{1}{\Delta t} < F(C_l^{(p)}). \quad (3.19)$$

402 This means that when the above is true the diagonal elements of A will be strictly larger then the sum
 403 of off-diagonals. Therefore we have diagonal dominance. \square

404 Since we have A positive definite, symmetric, and diagonally dominate we know that A is an M-
 405 matrix. This is important because this ensures that if M^k is non-negative we have that $M^{(p)}$ is also
 406 non-negative.

407 For solving (3.12), the equation can be rearranged into a quadratic form, substituting in $G(C)$ from
 408 (2.18)

$$409 \quad (C^{(p+1)})^2 + \left(\kappa - C^k + \frac{\Delta t}{2}\gamma M^{(p+1)} + \frac{\Delta t}{2}\frac{\gamma C^k M^k}{\kappa + C^k}\right) C^{(p+1)} + \left(-\kappa C^k + \frac{\Delta t}{2}\frac{\gamma \kappa C^k M^k}{\kappa + C^k}\right) = 0. \quad (3.20)$$

410 Using the quadratic equation results in,

$$411 \quad C^{(p+1)} = \frac{-b \pm \sqrt{b^2 - 4c}}{2} \quad (3.21)$$

412 for which,

$$\begin{aligned} b &= \kappa - C^k + \frac{\Delta t}{2} \gamma M^{(p+1)} + \frac{\Delta t}{2} \frac{\gamma C^k M^k}{\kappa + C^k} \\ c &= -\kappa C^k + \frac{\Delta t}{2} \frac{\gamma \kappa C^k M^k}{\kappa + C^k} \end{aligned} \quad (3.22)$$

414 Unless $b^2 - 4c = 0$, we have two different solutions to $C^{(p+1)}$. The problem with that is that if both
 415 solutions are positive we have two valid values to be used. Here, we can show that there will always
 416 be only one positive solution.

417 **Proposition 3.2.3.** *The quadratic equation defined as (3.20) will always have one positive solution
 418 and one negative solution for non-zero parameter choices.*

419 *Proof.* Rearranging (3.20) so that all the Δt terms are on the right-hand-side, we get

$$(C^{(p+1)})^2 + (\kappa - C^k) C^{(p+1)} - \kappa C^k = \left(\frac{\gamma C^k M^k}{2(\kappa + C^k)} - \left(\frac{\gamma M^{(p+1)}}{2} - \frac{\gamma C^k M^k}{2(\kappa + C^k)} \right) C^{(p+1)} \right) \Delta t. \quad (3.23)$$

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

422 We analyze both the left-hand-side and right-hand-side independently by letting $f_l = (C^{(p+1)})^2 +$
 423 $(\kappa - C^k) C^{(p+1)} - \kappa C^k$ and $f_r = (\bar{b} - \bar{a} C^{(p+1)}) \Delta t$. f_l is a quadratic equation with positive concavity
 424 everywhere and $C^{(p+1)}$ -intercept at $-\kappa C^k < 0$. f_r is a line with a slope opposite to the sign of \bar{a} and
 425 has $C^{(p+1)}$ -intercept at $\bar{b} \Delta t > 0$. Refer to Figure (3.1) for a visual of f_l and f_r .

426 Consider the case $\bar{a} < 0$. This means that f_r has a positive slope. We can show that f_r must intersect
 427 with f_l twice, one at $C^{(p+1)} < 0$ and another at $C^{(p+1)} > 0$.

428 For the positive intersection we can consider a point $c > 0$. Because $f_l = O((C^{(p+1)})^2)$, using Big-O
 429 notation, and $f_r = O(C^{(p+1)})$ we know that there exists a c for which $f_l > f_r$ for all $C^{(p+1)} > c$.
 430 Since we trivially have $f_r(0) > f_l(0)$ we can use the intermediate value theorem to determine that
 431 there must exist a intersection between f_l and f_r . This intersection is the point where $f_l = f_r$ and
 432 satisfies our equation resulting in a solution for $C^{(p+1)} > 0$. For the negative intersection we know
 433 that $f_r(0) > f_l(0)$ and f_r is monotonically increasing with $C^{(p+1)}$ and f_l is monotonically decreasing

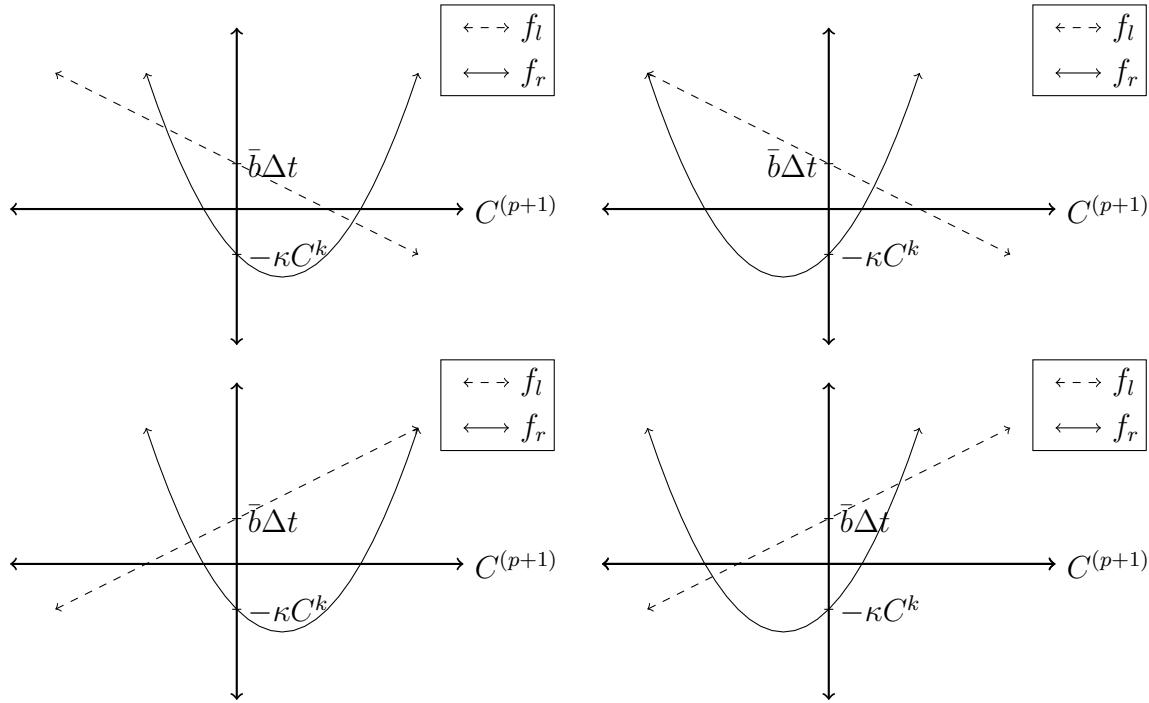


Figure 3.1: Graph of $f_l = (C^{(p+1)})^2 + (\kappa - C^k) C^{(p+1)} - \kappa C^k$ and $f_r = (\bar{b} - \bar{a} C^{(p+1)}) \Delta t$ for all four possible cases. Notice that because $-\kappa C^k < 0$ and $\bar{b} \Delta t > 0$ for all realistic parameter values the two functions will always intersect in the positive $C^{(p+1)}$ region. The top left graph is for $\bar{a} > 0$ and $\kappa - C^k < 0$. The top right graph is for $\bar{a} > 0$ and $\kappa - C^k > 0$. The bottom left graph is for $\bar{a} < 0$ and $\kappa - C^k < 0$. The bottom right graph is for $\bar{a} < 0$ and $\kappa - C^k > 0$.

434 for $C^{(p+1)} < 0$. This means that there must exist a point where the two functions intersect and thus
 435 we have a solution for $C^{(p+1)} < 0$.

436 The other case of $\bar{a} < 0$ follows a similar argument to the previous and also results in one positive
 437 solution and one negative solution for (3.20).

438 □

439 To determine which branch of (3.21) to use, a physical situation is used. Specifically the case where
 440 there exist no biomass, $M = 0$.

441 The expected outcome is that no substrate is consumed and thus the substrate concentration will
 442 remain constant as a function of t . When the equations in (3.22) are evaluated at $M = 0$, the result it,

443
$$b = \kappa - C^k, \quad c = -\kappa C^k, \quad (3.24)$$

⁴⁴⁴ which can be used to evaluate (3.21) as,

$$\begin{aligned}
 C^{(p+1)} &= \frac{-(\kappa - C^k) \pm \sqrt{(\kappa - C^k)^2 - 4(-\kappa C^k)}}{2} \\
 &= \frac{1}{2} \left(C^k - \kappa \pm \sqrt{\kappa^2 + 2\kappa C^k + (C^k)^2} \right) \\
 &= \frac{1}{2} (C^k - \kappa \pm (\kappa + C^k)).
 \end{aligned} \tag{3.25}$$

⁴⁴⁶ Now, if the positive branch is used the above equation evaluates to $C^{(p+1)} = C^k$. This means that be-
⁴⁴⁷ tween any two distinct times, the substrate concentration will remain constants, which was expected.
⁴⁴⁸ To further this confirmation, the negative branch results in $C^{(p+1)} = -\kappa$, a non-positive substrate
⁴⁴⁹ concentration, which is not physically relavent.

$$C^{(p+1)} = \frac{-b + \sqrt{b^2 - 4c}}{2} \tag{3.26}$$

⁴⁵¹ where b and c are defined in (3.22).

⁴⁵² Now that computable solutions for M and C at a single time step have been found, an algorithm to
⁴⁵³ solve for the next time step can be established. Algorithm 1 shows the organization of solving (3.12 -
3.11).

Data: M^k, C^k are vectors with values from the previous timestep and $p = 0$.

begin

```

    Let  $M^{(0)} = M^k$  and  $C^{(0)} = C^k$ ;
    while Convergence is not acheived do
        Solve  $A^{(p)}M^{(p+1)} = b^{(p)}$ ;
        Solve  $C^{(p+1)} = \frac{1}{2} (b \pm \sqrt{b^2 - 4c})$ ;
        Check convergence;
        Let  $C^{(p)} = C_{(p+1)}$ ;
        Let  $M^{(p)} = M_{(p+1)}$ ;
        Let  $p = p + 1$ ;
    end
    Let  $M^{(k+1)} = M^{(P)}$  and  $C^{(k+1)} = C^{(P)}$ ;
end

```

Algorithm 1: Algorithm for the fully-implicit solving of (2.16)

455 Note that Algorithm 1 actually describes both a fully- and semi- implicit method for solving (2.16).
 456 If $P = 1$ then only a single iteration of the algorithm is applied, which correlates to a semi-implicit
 457 method would behave. This can be produced by selecting a sufficiently large enough tolerance so that
 458 the convergence check is always resolved after the first iteration. The resulting semi-implicit method
 459 is effectively the same as one described in Sirca and Horvat (2012).

460 **3.3 Computational Setup**

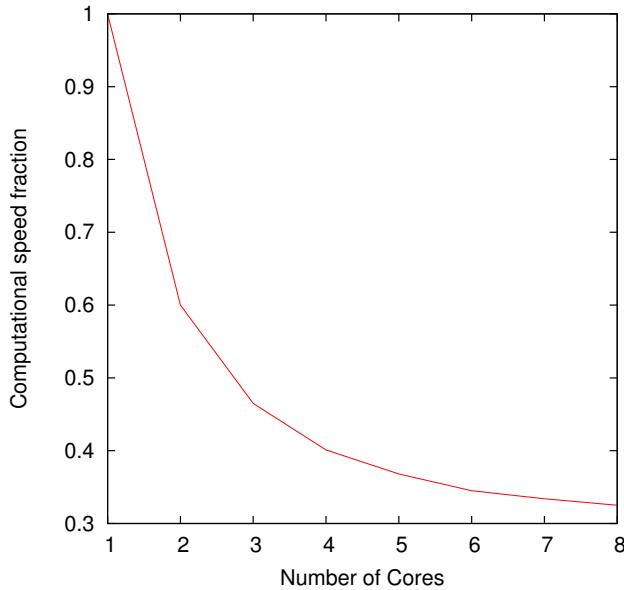


Figure 3.2: Computation speed of simulations ran with different number of cores. The y -axis represents the fraction of time each core has in comparison to the 1-core result. Same simulation setup as the travelling wave simulation. Ran until $t = 2$ with a grid of 2049×2049 .

461 The implementation of Algorithm 1 was done with Fortran. The system is stored in diagonal format,
 462 since A has five distinct diagonals.
 463 All the computations were run on a custom built workstation with an Intel Xeon CPU E5-2650 (1.2
 464 GHz, 20MB cache size) and 32 GB RAM under Red Hat Enterprise Linux Server release 6.5 (San-
 465 tiago). Running the computations with OpenMP, took advantage of 4 out of the 16 threads of the
 466 Intel Xeon CPU, with 2 threads to each core. The choice of 4 cores is because there is not enough
 467 computational gain from using extra, as shown in Figure 3.2. The selection of 4 threads is because
 468 the computation time decrease becomes less efficient after more than 4 threads. The GNU Fortran

469 compiler, version 4.4.7, was used for all computations; the compiler arguments were

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

471 3.4 Method Validation

472 With a defined method and computational setup we assess the behavior and accuracy of the method in
 473 a variety of simulations. An examination of a typical simulation will show if the expected behaviour is
 474 observed. A convergence analysis for the method can be done to confirm that solutions from different
 475 grid sizes approach a single solution as they become more precise. This convergence test will also
 476 show the thresholds for an accurate simulation result, to help reduce the computation times. Once
 477 the fully-implicit method has been tested, it can be compared against the semi-implicit method.

478 3.4.1 Basic Simulations

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

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

$$484 M = \begin{cases} -\left(\frac{h}{d^4}\right)x^4 + h & , \text{if } y \leq d \\ 0 & , \text{otherwise} \end{cases} \quad (3.27)$$

$$C = 1$$

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

486 The solution shown in Figure 3.3 shows that the 1D characteristic of the biomass stays at a later time.

487 Another characteristic to observe would be if a spherical initial condition remains spherical. Using

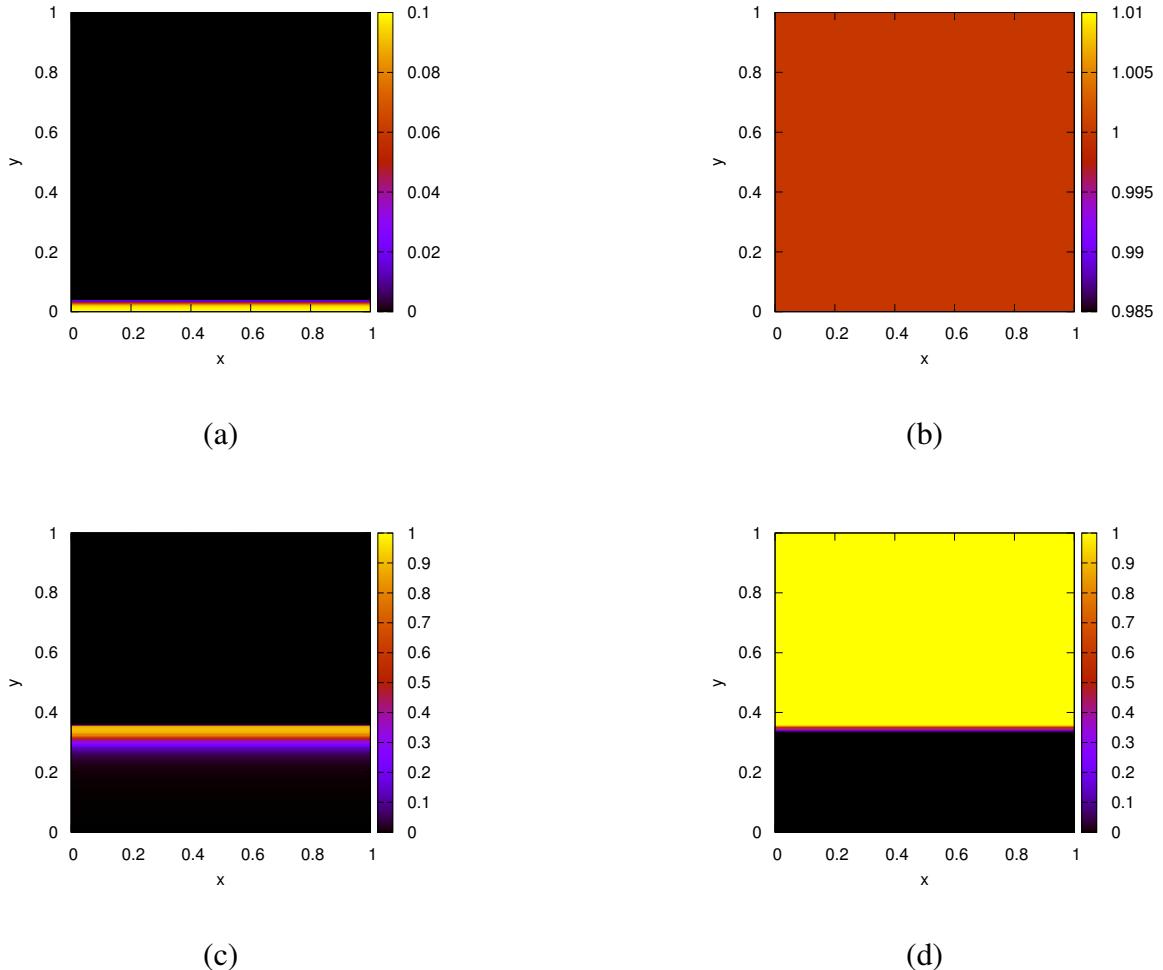


Figure 3.3: Solutions for (ac) M and (bd) C with 1D initial conditions defined in (3.27) at (ab) $t = 0$ and (cd) $t = 40$. Computed with a 1025×1025 grid and a timestep of $\Delta t = 10^{-3}$.

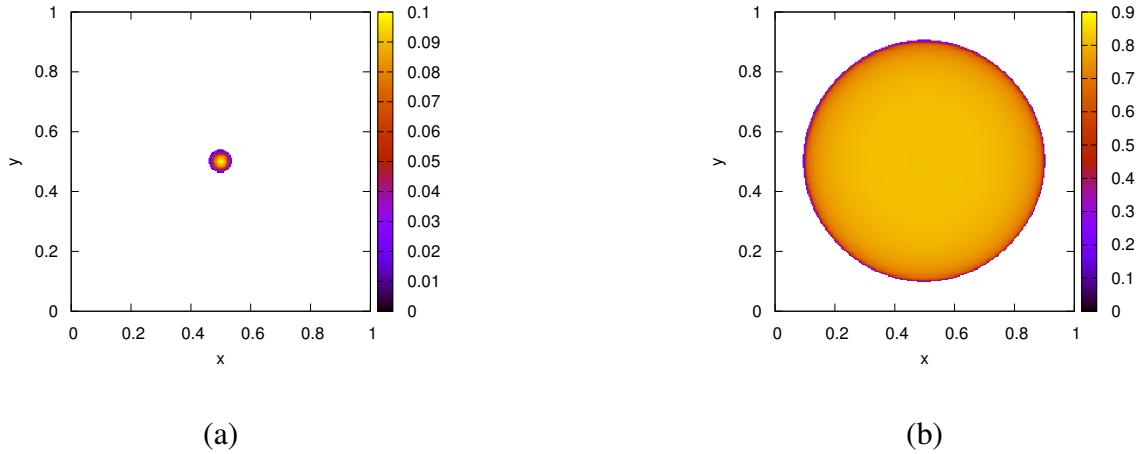


Figure 3.4: Solutions for M with spherical initial conditions defined by (3.28) at (a) $t = 0$ and (b) $t = 40$. Computed with a 1025×1025 grid and a timestep of $\Delta t = 10^{-3}$.

488 intial conditions for the biomass,

$$489 \quad M = \begin{cases} -\frac{h}{d^2} (x - 0.5)^2 + (y - 0.5)^2 + h & , \text{if } (x - 0.5)^2 + (y - 0.5)^2 < d^2 \\ 0 & , \text{otherwise} \end{cases}, \quad (3.28)$$

490 a test can be tried to see if the spherical nature of the solution is kept as time progresses. The solution
491 shown in Figure 3.4 shows that the spherical shape of the solution is maintained at later times.

492 Both Figure 3.3 and Figure 3.4 increase the confidence that the spatial discretization did not introduce
493 any loss of characteristics for the solutions.

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

500 The expected total biomass would be of the form $y_0 e^{at}$. This can be checked by tracking the total

501 biomass, now called $T_M(t)$, with the changed growth rate function, $F(C) = a$. The calculation of
 502 $T_M(t)$ can be done by,

$$503 \quad T_M(t) = \int_{\Omega} M(t) dx. \quad (3.29)$$

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

$$505 \quad T_M(t^k) \approx T_M^k = \frac{\sum_i^n \sum_j^m M_{i,j}^k}{nm}. \quad (3.30)$$

506 The simulation setup used will be analogous to that used for Figure 3.4. The one difference will be
 507 that the simulation here is ran for a longer time to allow the biomass to diffuse along the boundary,
 508 showing the boundary effects.

509 From Figure 3.5 we can see that the total biomass only differs between the computed value and the
 510 theoretical value by a relative error less than 0.003. The cases where the error becomes significant are
 511 from the region being completely filled with biomass, at which point diffusion is no longer possible.
 512 The error fluctuates violently here because of this. This suggests that the method does not introduce
 513 any significant sources or sinks of biomass at the boundary of the region.

514 3.4.2 Convergence Analysis

515 To validate the accuracy of the method, convergence analyses on the spatial discretizations will need
 516 to be made. Then the comparison between the semi- and fully-implicit method established in Algo-
 517 rithm 1 can be investigated. First, a metric must be formed to enable consistent comparisons between
 518 different simulation solutions. This metric will be referred to as the normed difference. Only M will
 519 be considered for the normed difference calculations. This is because C depends on M and including
 520 it does not qualitatively change the results.

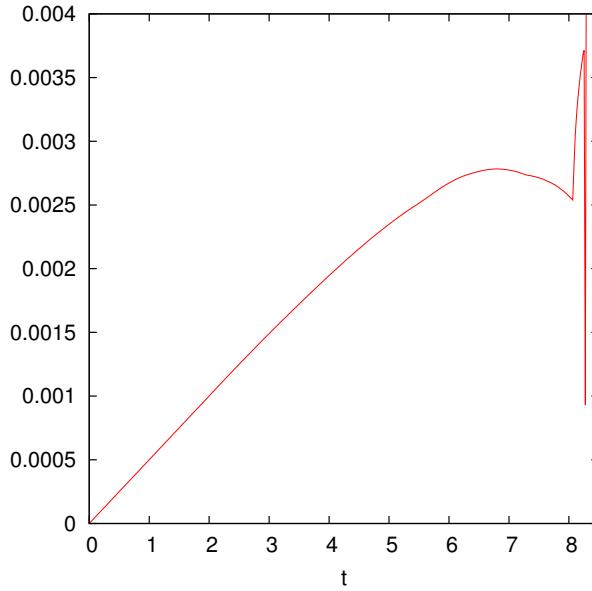


Figure 3.5: Plot of the relative error, $\frac{|f_1 - f_2|}{|f_2|}$, between the computed total biomass, $f_1 = T_M(t)$, and the theoretical total biomass, $f_2 = y_0 e^x$. The changes after $t = 8$ are from the biomass having completely filled the region Ω . This means that there is no physical space for the biomass to occupy and thus the growth slows down to a stop.

521 **3.4.2.1 Normed Difference Computations**

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

524
$$\epsilon_{sol} = \frac{\|u_1 - u_2\|}{\|u_2\|} \quad (3.31)$$

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

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

534 will be used:

$$535 \quad \ell_1 : \|u\|_1 = \frac{1}{nm} \sum_i^{nm} |u_i| \quad (3.32)$$

$$536 \quad 537 \quad \ell_2 : \|u\|_2 = \frac{1}{nm} \sqrt{\sum_i^{nm} (u_i)^2} \quad (3.33)$$

$$538 \quad 539 \quad \ell_\infty : \|u\|_\infty = \max_{i=1,\dots,nm} |u_i| \quad (3.34)$$

540 These different norms will all be used to create a broader understanding of the normed difference.
 541 This creates three distinct values for ϵ_{sol} , named ϵ_{ℓ_1} , ϵ_{ℓ_2} , and ϵ_{ℓ_∞} ; each named for the norm used
 542 during the computation. Note that these norms are for the vector for of the solution, throught the use
 543 of the grid-ordering $\pi(i, j)$.

544 3.4.2.2 Grid Size Convergence

545 To observe the validity of the method, a test on the convergence of solutions based on the spatial
 546 discritization is done. This will involve using the same simulation described in (3.27) due to the
 547 simplicity.

548 The convergence will be tracked with only two forms of ϵ_{sol} ; ϵ_1 and ϵ_2 . This is because the value
 549 of ϵ_∞ doesn't vary with the grid size, since the wave front has a steep interface and tends to lead to
 550 inconsistent changes in normed difference. Since the use of ϵ_∞ is not a suitable method for measuring
 551 the normed difference, the inconsistency does not suggest an invalidity with the method. Because of
 552 the difference in the number of grid points between different solutions, u_1 and u_2 , only the grid points
 553 in the coarser refinement will be used. This places a limitation on the selection of grid-sizes since
 554 there must be some grid points locations that are the same for two different chosen grid sizes. For
 555 this purpose, we define the function $s(n) = 2^n + 1$ for $n \in \mathbb{N}$ to be used as the grid size selection
 556 function. Now certain grid points will match without the use of linear interpolation, as illustrated in
 557 Figure 3.6.

558 Using the same simulation setup as was done in Figure (3.3), solutions resulting from different grid

	$s(n) = 2^n + 1$				
	1	2	3	4	5
$n = 2$	—	—	—	—	—
$n = 3$	—	—	—	—	—

Figure 3.6: Visualization in 1D to illustrate the choice of $s(n) = 2^n + 1$ instead of 2^n for the grid size selection. Here it can be seen that successive grid size selections using $s(n)$ line up on certain grid points and when using 2^n no grid points are equivalent.

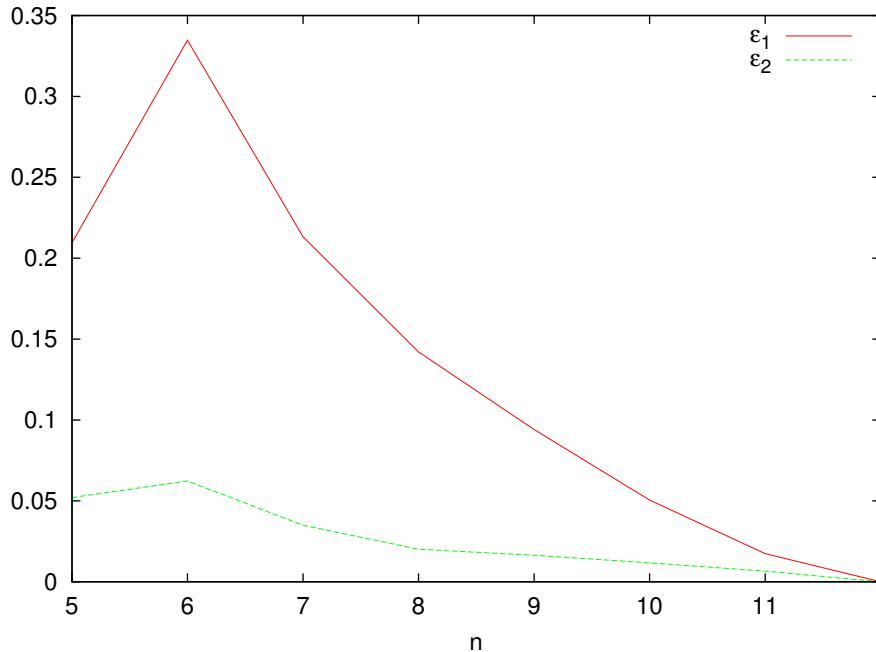


Figure 3.7: Plot showing the convergence of solutions based on changes in Δx . The computations are of ϵ_{ℓ_1} and ϵ_{ℓ_2} with grid-size following $s(n) = 2^n + 1$.

559 sizes based on $s(n)$ are computed for $n = 5, 6, \dots, 12$. In this case, when calculating $\epsilon_{sol} = \frac{|u_1 - u_2|}{|u_2|}$,
 560 we let u_1 be the grid size under investigation and u_2 be the solutions of the most refined gridsize,
 561 $n = 12$. This would show the converging solutions for smaller grid sizes because the change to the
 562 finest grid size will be monotonically decreasing.

563 The results from Figure 3.7 show that the solutions converge as the grid size become refined with the
 564 grid size. The nonmonotonicity for $n = 5$ is because the grid has become too large for predictable cal-
 565 culations, which is an acceptable error since such a coarse grid would never be used for computational
 566 simulations.

3.5 Comparison of Semi-implicit and Fully-implicit Method

We are now in a position to compare the fully-implicit time-integration scheme that we introduced here with the semi-implicit time-integration that has been used for problems with a diffusive substrate in the literature. Recall that the semi-implicit method is a special case of the fully-implicit method if the non-linear iteration is stopped after one step. This can be forcibly achieved, for example, by choosing a very large tolerance threshold for the non-linear iteration.

The simulation used is the same as described in (3.3). The comparison will be on multiple metrics: the average number of iterations of Algorithm 1, the value of ϵ_1 and ϵ_2 , the computation time of the simulation, and the height of the wave peak.

The average number of iterations are tracked so that an idea of the extra work can be formed. This average is based on the average number of iteration of the fully-implicit method from $t = 0$ to the current t . This value is used since it represents the number of iterations in a way that is easy to read and understand. As the tolerance decreases the amount of iterations the algorithm must perform will increase, the degree of increase will help relate the amount of work.

The value of ϵ_1 and ϵ_2 act as a measure of accuracy. Here, these values correspond to the difference between a pair of solutions, u_1 and u_2 . The choice of u_2 here is the semi-implicit methods solution. This results in a relative difference from the semi-implicit method and would show the change in solution as the tolerance decreases. Each row of Table 3.1 refers to the u_1 values used in the comparison. Each difference was taken at the last timestep.

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

The location of the wave peak is a tracked quality of the solution that reveals how consistent the

592 results are. The wave peak is described here as the maximum value of the solution at the final timestep
 593 calculated. The ultimate goal is that the simulation solutions be converging towards the exact solution.
 594 To see this here the x -coordinate of the wave peak is tracked as well as the height of the wave peak.

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

Tol.	Avg. Iter.	ϵ_1	ϵ_2	Time	Wave Height
1.0e-0	1.0000	0.00000000000000	0.00000000000000	12.1830	0.96366123
1.0e-1	1.0000	0.00000000000000	0.00000000000000	12.2080	0.96366123
1.0e-2	1.0000	0.0000000000428	0.0000000000167	12.3379	0.96366123
1.0e-3	1.0000	0.0000000000428	0.0000000000167	12.2310	0.96366123
1.0e-4	1.0000	0.0000000001098	0.000000000329	12.3200	0.96366123
1.0e-5	1.9650	0.002573969658	0.001066499658	18.9869	0.96391491
1.0e-6	2.0000	0.002574057907	0.001066505860	19.0910	0.96391479
1.0e-7	2.0018	0.002573959764	0.001066498736	19.0940	0.96391492
1.0e-8	2.5856	0.002577916965	0.001066781565	20.5169	0.96390966
1.0e-9	2.9012	0.002577461054	0.001066759099	21.3080	0.96390979
1.0e-10	3.2278	0.002581334868	0.001067029069	22.2280	0.96390490
1.0e-11	16.0990	0.002632955188	0.001070709373	57.5589	0.96383105
1.0e-12	36.3184	0.002647234923	0.001071748013	113.9940	0.96380854
1.0e-13	57.6812	0.002648733848	0.001071857564	173.8489	0.96380614

Table 3.1: Results from running simulations with different Tol.

596 There are a number of observations that can be made from these results.

- 597 • A positive relationship between computation time and average number of iterations exists.
- 598 • There is no significant difference between solutions unless the tolerance is set high enough to
 599 force multiple iterations. So the semi-implicit method results in a tolerance between 10^{-4} and
 600 10^{-5} since any tolerance forced below that does not require additional iterations.
- 601 • The differences in Wave Height are a result of additional iterations and monotonically approach
 602 a specific value as the tolerance becomes smaller.
- 603 • The greatest gain in accuracy while weighing the increased computation time is from a tolerance
 604 around 10^{-5} at which point only one extra iteration is completed.

605 **Chapter 4**

606 **Simulation Results**

607 **4.1 Typical Simulation**

608 A typical simulation refers to the parameter values and the choice of initial condition. It will show the
609 behaviour of the system under normal circumstances and help reveal the interesting characteristics.

610 The typical initial condition attempts to emulate the biological situation of biomass growing inwards
611 on a sheet of cellulose. This will show how the biomass moves and how two separate masses interact
612 in a collision. The initial condition used will initialize a number of random spherical inoculation
613 points near the $y = 0$ and $y = 1$ axis. We let (x_r, y_r) be the random point used as the center for
614 inoculation. To separate the inoculation points we have $x_r \in [0, 1]$ and $y_r \in [0, 0.1] \cup [0.9, 1]$. The
615 number of inoculation points are the same for both the $y = 0$ region and $y = 1$ region. Multiple
616 inoculation points combine additively. Each spherical inoculation point is computed as,

617
$$M(0, x, y) = \frac{-h}{d^2} ((x - x_r)^2 + (y - y_r)^2) + h, \quad M \geq 0. \quad (4.1)$$

618 Note that $M(0, x, y)$ is for points within circles of radius d centered at (x_r, y_r) , otherwise $M(0, x, y) =$
619 0. For the substratum we have $C(0, x, y) = 1$ everywhere.

620 The choice of parameter value is based on the default values given in Table A.1. There are 40 innoc-

ulation points of both sides, totalling 80. The fully-implicit method is used here with $tol = 10^{-8}$.

4.1.1 Biomass Ratio

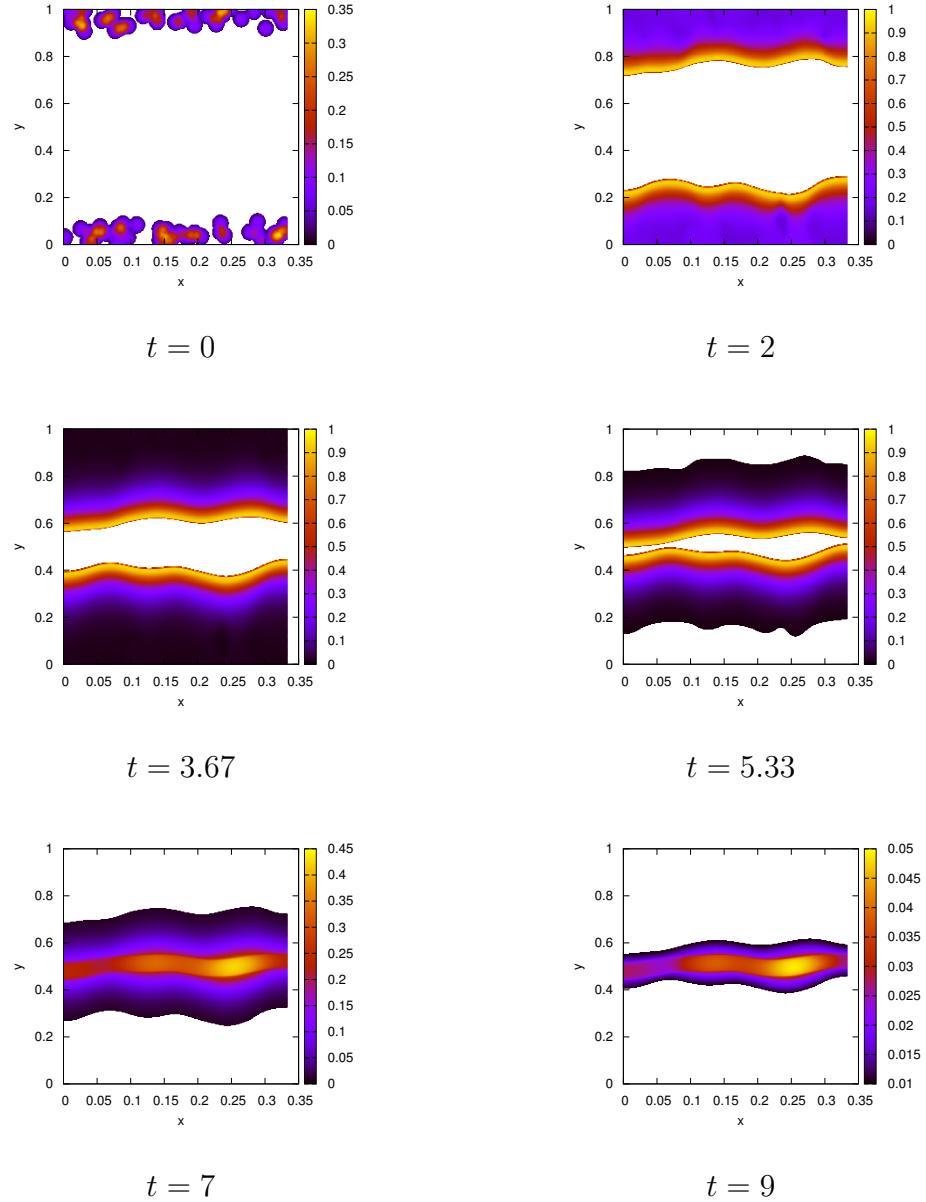


Figure 4.1: A graph showing the M solution of a typical simulation at different time steps. The initial condition is 80 random spherical innoculation points evenly divided between each of the $y = 0$ and $y = 1$ sides. A 513×513 grid was used.

Figure 4.1 shows the time evolution of M for the simulation. Here, the random innoculations on both sides of region progegate towards each other and eventually combine at the center. By looking at

625 $t = 2$, $t = 3.67$, and $t = 5.33$ it appears as though the wave front is moving with a constant shape and
 626 at a constant speed. This suggest that there may be the existence of a travelling wave solution.

627 One important feature to notice is that the time evolution in Figure 4.1 matches the conceptual model
 628 proposed in Dumitrache et al. (2015). This model can be seen in Figure 1.1. The different stages of
 629 the conceptual model can be observed in our simulation results:

- 630 • Stage I: $t = 2$ and $t = 3.67$ show the biomass growing towards the center of the sheet, which is
 631 the center white area.
- 632 • Stage II/III: $t = 5.33$ shows the consumed substate region as the outer white.
- 633 • Stage IV: Not shown. Only occurs at the moment when the two band first collide and the
 634 biomass concetration at that point still remains at the actual carrying capacity.
- 635 • Stage V: $t = 7$ and $t = 9$ show the combined center band, now at a biomass concentration
 636 lower then the actual carrying capacity.

637 **4.1.2 CO_2 Production**

638 Some important quantities to track are the total amount of biomass, M , and substrate, C . These values
 639 will be called $T_M(t)$ and $T_C(t)$ to represent the total biomass and total substrate, respectively. The
 640 computation for these values can be done by integrating over the region, Ω :

641
$$T_M(t) = \int_{\Omega} M dA, \quad T_C(t) = \int_{\Omega} C dA \quad (4.2)$$

642 These values can be seen in Figure 4.3 (bd) for $T_M(t)$ and (c) for $T_C(t)$.

643 Since *C. Thermocellum* produces CO_2 as the substrate is consumed, we can track the production of
 644 CO_2 . Following the idea from Dumitrache (2014), we can equate the change in production of CO_2

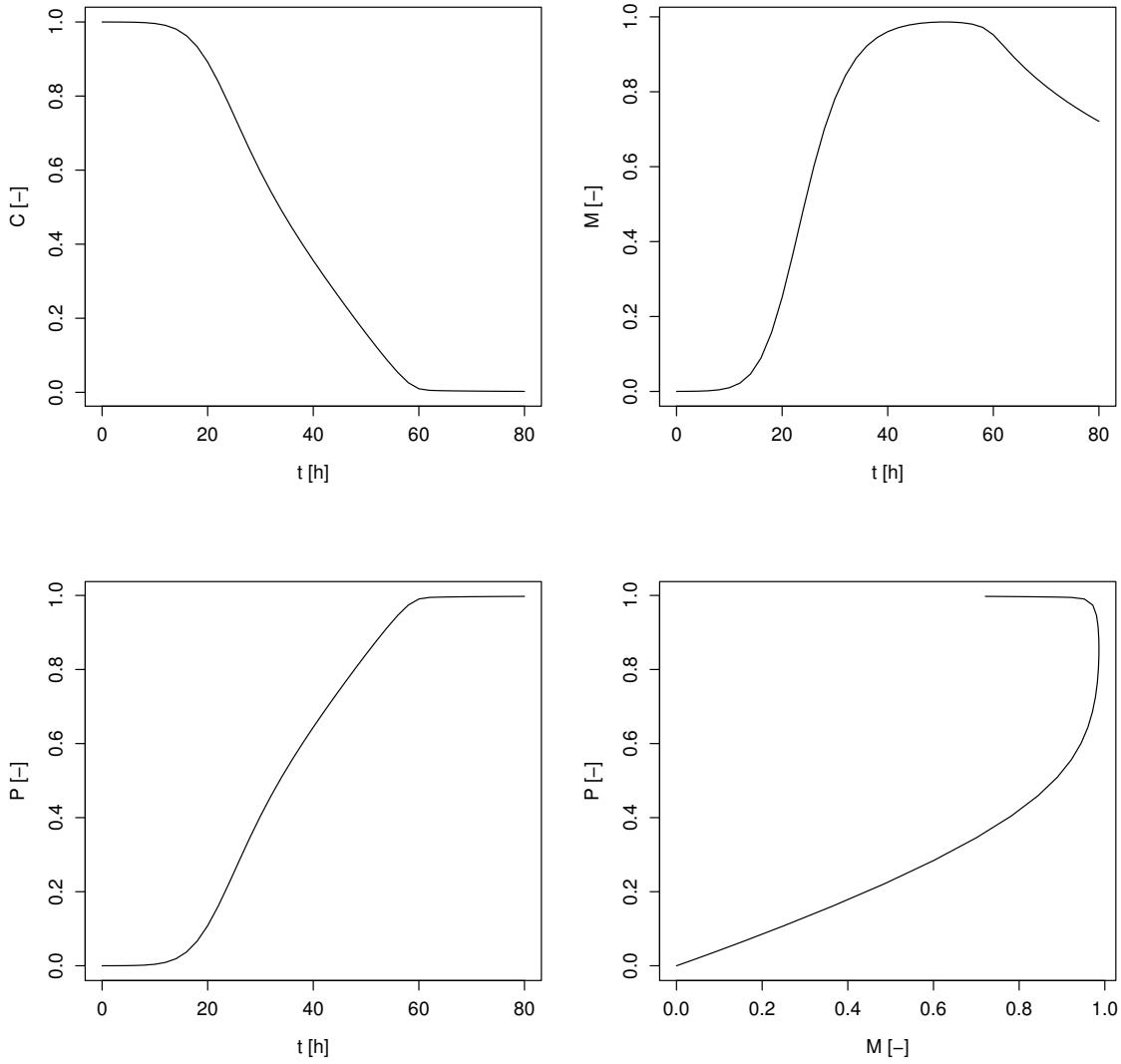


Figure 4.2: A typical model simulation from the simple ODE model. Shown are substrate concentration C (top left, normalised), effective sessile biomass M (top right, relative to the ideal carrying capacity M_∞), and CO_2 product P (bottom left, in moles) as functions of time t ;. Also shown is the product P vs sessile biomass M (bottom right, in moles). Figure originally from Dumitrache et al. (2015).

as time changes by the following equation:

$$p_t = \rho G(C)M. \quad (4.3)$$

645

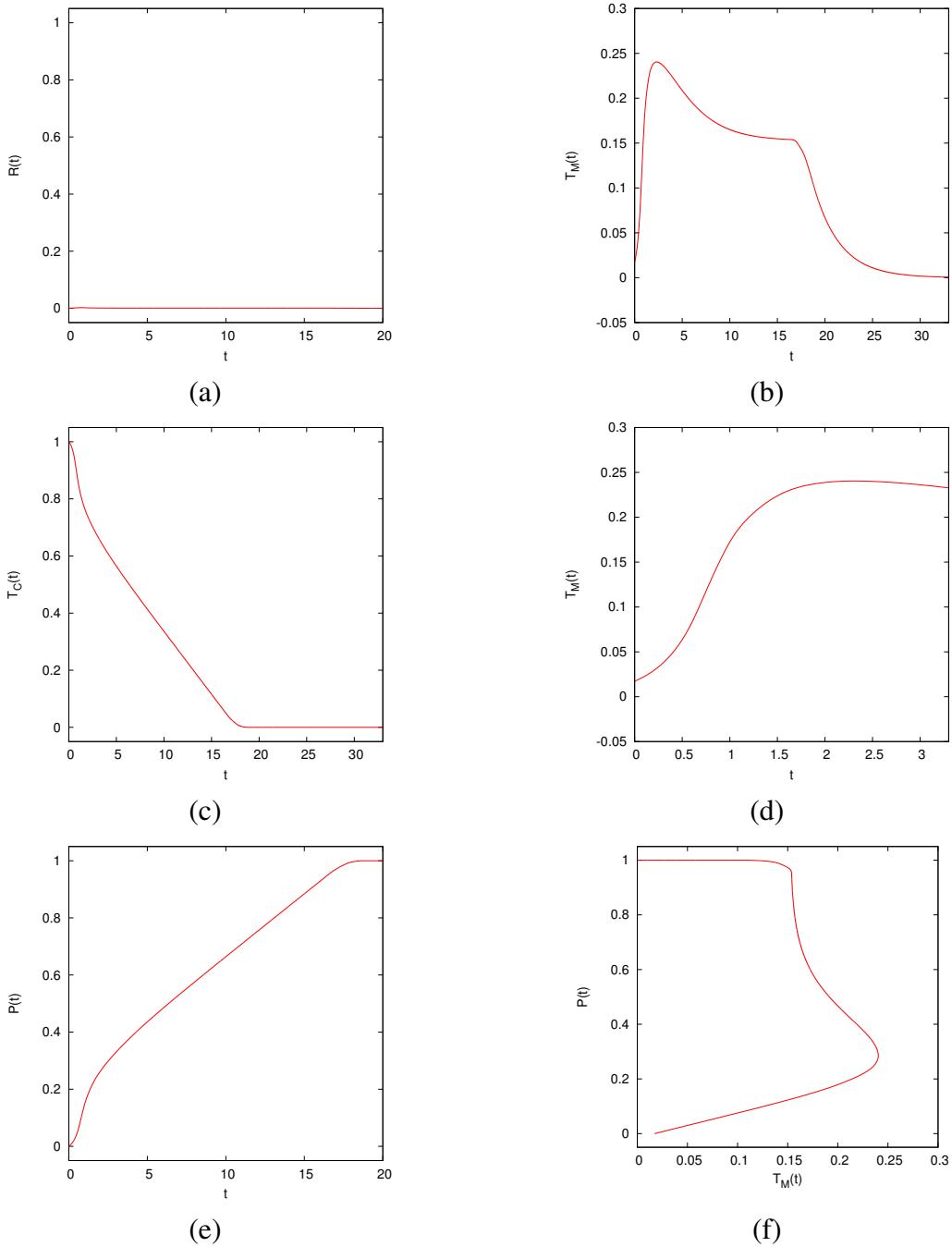


Figure 4.3: Total value of certain qualities from the typical simulation. Here we have: (a) $\mathcal{R}(t)$, the rate of CO_2 production, (b) total M as a function of time, (c) total C as a function of time, (d) total M as a function of time zoomed in from $t = 0$ to $t = 3$, (e) $\mathcal{P}(t)$, the total CO_2 produced, (f) $\mathcal{P}(t)$ as a function of total M . All the graphs are from the same simulation with initial condition of 40 random spherical inoculation points along the $y = 0$ side of the region and another 40 on $y = 1$. A grid of 257×257 was used for this graph. Default parameter set (Appendix A) was used except for $\delta = 10^{-8}$.

⁶⁴⁷ To get the amount of CO_2 produced at a specific time we get,

$$\mathcal{R}(t) = \int_{\Omega} p_t dA = \int_{\Omega} \rho G(C) M dA. \quad (4.4)$$

649 From this we can get the more useful value, the total CO_2 produced until this point.

650

$$\mathcal{P}(t) = \int_0^t \mathcal{R}(s)ds. \quad (4.5)$$

651 The CO_2 amount is calculated by letting $\rho = 1$ and using the numerically computed values for $G(C)M$
 652 as a measure. For the same simulation as Figure 4.1, the CO_2 information can be seen in Figure 4.3
 653 (a e).

654 The results from Figure 4.3 (c d e f) seems to match the results from the ordinary differential equation
 655 model proposed in Dumitache (2014). Their results can be seen in Figure 4.2. It is important to note
 656 that in our system $T_M = 1$ means that Ω is completely filled with biomass. However, in Dumitache
 657 et al. (2015) they scaled the biomass to the ideal carrying capacity of biomass, i.e. they have $T_M = 1$
 658 when all the biomass is in stage II or III. The overall result from this experiment is that the spatial
 659 two dimension model confirms the conceptual model from Dumitache et al. (2015) based on which
 660 the reactor-scale model was formulated. The reactor-scale model consolidated the spatial effects into
 661 the carrying capacity of the growth and yet still managed to agree with the results of the actual spatial
 662 model.

663 **4.2 Travelling Wave Analysis**

664 **4.2.1 Spatial Simplification**

665 To simplify the travelling wave analysis we reduce the spatial dimensions to that of a 1D problem.
 666 This can be done if initial conditions that are homogenous with respect to y are chosen. The purpose
 667 of this spatial simplification is that this will speed up the computations considerable. It will also make
 668 visualizations easier as certain figures would become too cluttered in 2D. What is done here is more
 669 of a pseudo-reduction of dimensions. By reducing the grids from an $n \times m$ grid to an $n \times 4$ grid
 670 we have changed the way the problem size scales with finer grids. The problem is still 2D, just now
 671 one dimension has been reduced to only 4 grid points of accuracy instead of m points. This does not

672 effect the final result since we only apply this change to problems with appropriate initial conditions.
 673 These initial conditions are homogenous in the y direction and thus we do not have any fluctuation
 674 between y values for a given x value.

675 One main benefit of changing the grid from $n \times m$ to $n \times 4$ is that the growth of the problem with
 676 respect to the resolution of the grid is reduced dramatically. This changes the problem from a $O(n^2)$
 677 problem to a $O(n)$. Using the travelling wave 1D initial conditions one simulation is computed with
 678 a 513×513 grid, seen at Figure 4.4, and another with a 513×4 grid, seen at Figure 4.6.

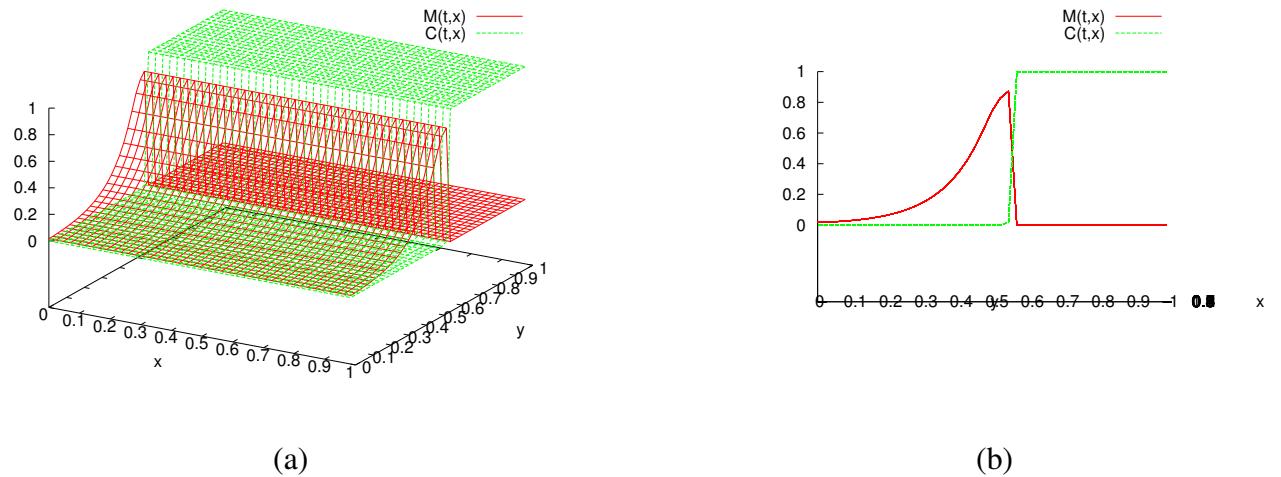


Figure 4.4: Graph of (a) 3D view of $M(t, x, y)$ and $C(t, x, y)$, (b) Side profile view of $M(t, x, y)$ and $C(t, x, y)$ at $t = 40$.

679 Before any changes to the grid can be made, it must be confirmed that fluctuations are sufficiently
 680 small. To this end, the standard deviation is used as a measure. The standard deviation is calculated
 681 along the y -direction for each x value. This gives a numerical quantity for the measure of dispersal
 682 each y value has with another. Here, we use the sample standard deviation for the sole reason that this
 683 single simulation does not represent its own population. Initially, at $t = 0$ the standard deviation is 0
 684 everywhere (DATA NOT SHOWN). At $t = 40$, Figure 4.5 show the standard of each y value. After
 685 many timesteps have passed the amount of spread is always less than 10^{-14} , which is an acceptable
 686 degree of consistency. Note that the main inconsistency is at the wave front, around $x = 5.75$, which
 687 is mainly because of the sharp change in values.

688 When simulations are computed with a $n \times 4$ grid, they are still 2D problems. With regards to

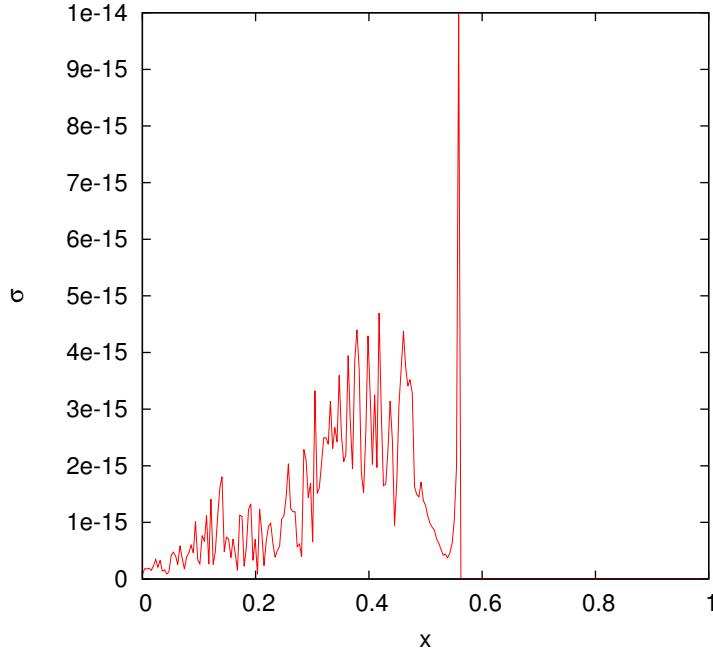


Figure 4.5: The standard deviation at the same time as the above graphs

689 visualizations, side profiles could be used on these solutions to present psuedo-1D visualization but
 690 this is not ideal. To visualize the solutions in true 1D the we use \bar{M} and \bar{C} as averaged values of the
 691 solutions along the y-axis. This is computed after the solution has been determined and is independent
 692 of the actual compuations for M and C. So by taking the average of the points along the y-axis we
 693 can get a 2D plot as seen in Figure 4.6.

694 This means that the system can be reduced to a 1D problem. With initial conditions that are homoge-
 695 nous with respect to y, we can greatly reduce the accuracy in the one axis. Once the y-axis reduced,
 696 we can also ignore it for visualizations, only using the x-z axis and plotting the values of \bar{M} and \bar{C} .

697 4.2.2 Travelling Wave Solution

698 Classical travelling wave solutions are solutions that propagate with an *a priori* unknown constant
 699 speed without any change in shape. This means that the solutions can be defined as

700
$$M(t, \tilde{x}) = M(\tilde{x} - ct) \quad (4.6)$$

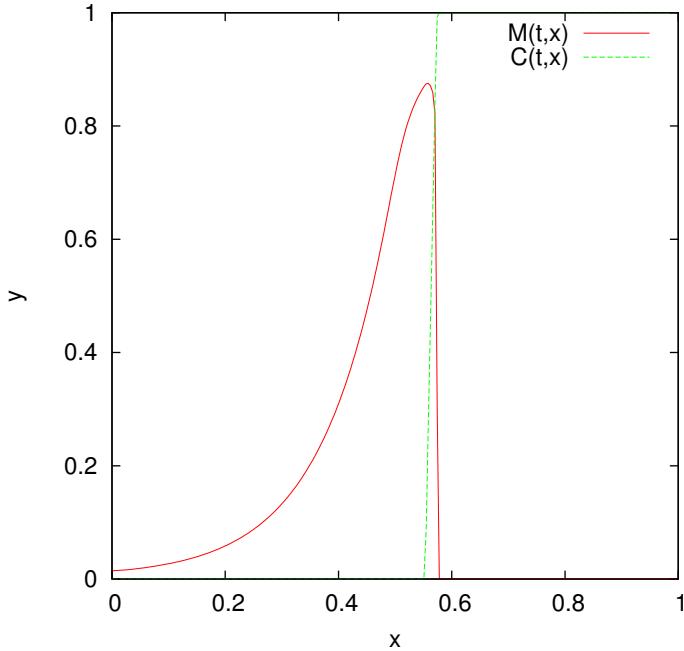


Figure 4.6: Graph of $M(2,y)$ and $C(2,y)$, now reduced to a 2D plot.

701 Figure 4.7 shows the time evolution of the single time snapshot from Figure 4.6. Given the above
 702 definition and by looking at the consistent appearance of the solution, it suggests that it is a travelling
 703 wave. It is clear here that the shape of the solution is consistent enough to suggest the existence of a
 704 travelling wave solution.

705 The existence of a travelling wave solution for this simulation can be confirmed if the solution $M(x, t)$
 706 can be shown as $M(x - ct)$, where c is the *a priori* unknown wavespeed. Visually, multiple timesteps
 707 horizontally translated onto each other would show this. If the horizontal translations are all multiples
 708 of the same number then we have shown that a constant speed exists. If the shape of all the timesteps
 709 match then a constant shape then strong evidence that a constant shape exists would be shown. We
 710 can numerically approximate the value for c by looking at how fast the peak of the wave travels. The
 711 location of the wave peak is the x coordinate that corresponds to the largest M value. Recall that we
 712 are dealing with a pseudo-1D problem, so there does not need to be any consideration for an (x, y)
 713 coordinate. For this case, we used the GNUPLOT software to fit a linear model, $f(x) = mx + b$, to
 714 the last half of the wave peaks path, seen in Figure 4.8. The last half of the values were used instead

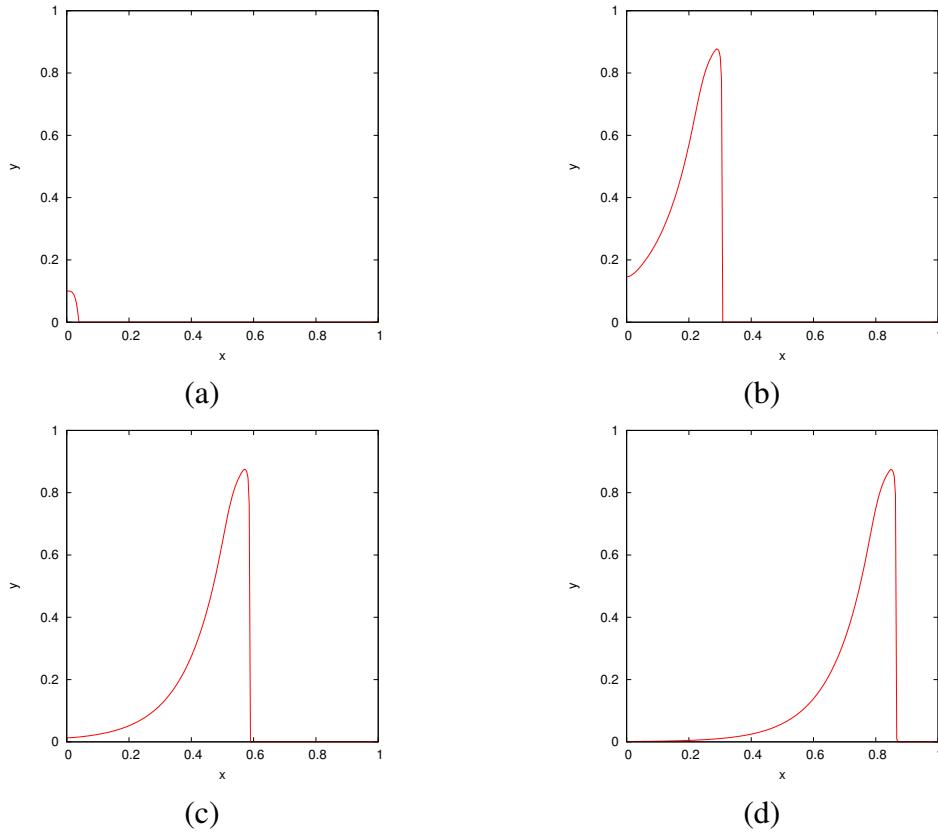


Figure 4.7: Solutions of $M(x, t)$ and $C(x, t)$ at (a) $t = 0$, (b) $t = 20$, (c) $= 40$, (d) $= 60$. This was run on a 513×4 grid.

715 of the whole set of values because only for the former do we have a fully formed travelling wave. The
 716 value of m in $f(x)$ is the approximation for the wavespeed, c .

717 With an approximation for c , the solutions of Figure 4.7 can be represented as $M(x - c(t_0 - t_n))$,
 718 where $t_0 = 60$ is a reference point for the other timesteps. The values of t_n are the times for the other
 719 solutions. By translating along the x -axis multipl solution profiles can be superimposed, as seen in
 720 Figure 4.9. The shape of each timestep is very similar throughout, only differing slightly at the tail.

721 Based on the above evidence, we can say that a travelling wave solution has been shown to exist for a
 722 single initial condition and particular set of parameters. This leads to two logical extensions, looking
 723 at the stability of the travelling wave solution based on initial condition and investigating the effect
 724 the parameters have on the travelling wave solution.

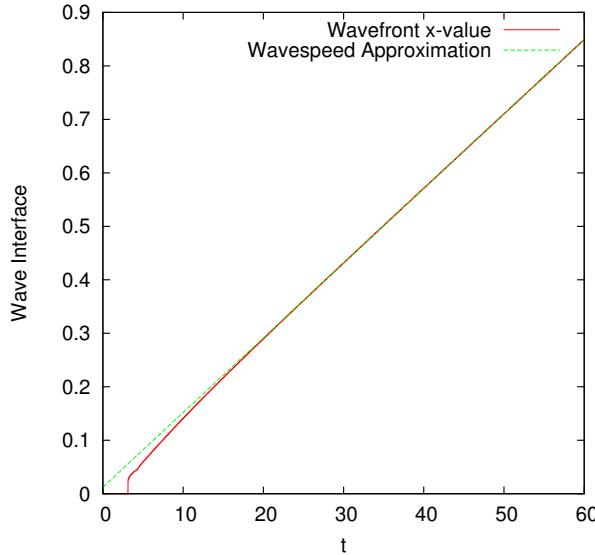


Figure 4.8: The x location of the wave peak as a function of t . The red line is the wave peak location extracted from the simulation results. The green line is the function $f(x) = cx + b$ with c as the wavespeed, found by fitting the model to the second half of x values. The simulation results used here are from the solution shown in the previous Figure.

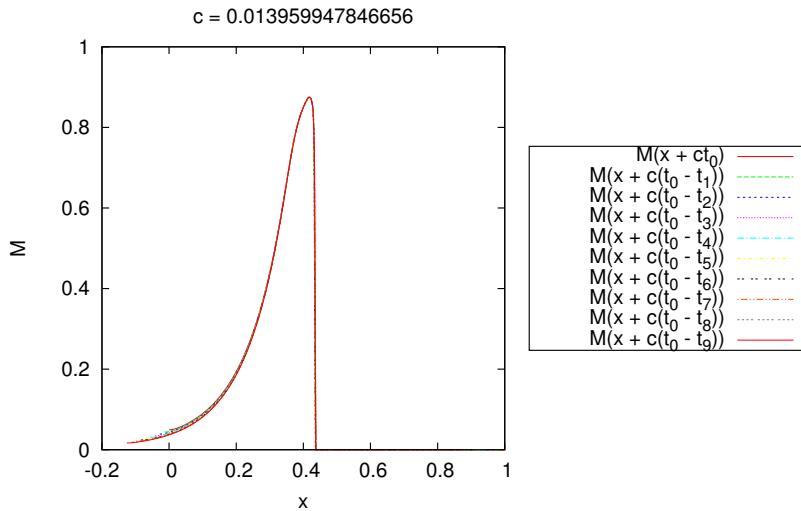


Figure 4.9: Solutions of M that are represented as $M(x - ct)$ *a priori*. The multiple timesteps are translated on top of another by horizontal movements of $c(t_0 - t_n)$ for each timestep.

725 4.2.3 Travelling Wave Stability

726 Based on the previous example, there seems to exist a travelling wave solution. The next step is
 727 looking at how different initial conditions could still result in a travelling wave solution. For this

728 we specifically look at the stability of the solution, does it attract nearby solution into becoming a
 729 travelling wave solution or is it only for specific cases that one results. This will help confirm that the
 730 existence of the travelling wave solution is not depended on the single choice of initial condition.

731 To test this we take an initial condition that is not inherently one dimensional and see if it approaches
 732 to the one dimension property. The choice of IC is to have multiple random spherical inoculation
 733 points along the $y = 0$ side of the region. Specifically, we use (x_r, y_r) to represent the center of each
 734 random inoculation point. Here $x_r \in \mathcal{R}$ and $y_r \in [0, 0.1]$.

735 The equation used for each random spherical inoculation point is,

$$736 \quad M = \frac{-h}{d^2} ((x - x_r)^2 + (y - y_r)^2) + h, \quad M \geq 0. \quad (4.7)$$

737 Random inoculation points add to each other if they overlap. After all the inoculation points have
 738 been generated, every value is divide by the total amount of biomass. This lets the initial condition
 739 become a representation for the distribution of random inoculation points in terms of the total amount
 740 generated. A time evolution of the simulation with the above initial condition can be seen in Figure
 741 4.10. Here it can be observed that the solution M appears to slowly converge to a 1D problem. This
 742 cannot be fully seen since the wave propagation reaches the end of the region before it can become
 743 fully one dimensional.

744 We can quantitatively see the behaviour of this convergence by calculating the measure of spread at
 745 the wave front. This can be achieved by calculating the standard deviation of y coordinate for each
 746 x coordinate. By tracking the largest y value with a non-zero M for each x value we can generate a
 747 sample data set of the wavefront. The wavefront is used instead of other points of interest, such as
 748 the wave peak, because it is the most consistent of characteristics that can be easily tracked. As seen
 749 in Figure 4.5, the wave peak had the largest spread among all other values.

750 Taking the sample standard deviation of this set results in the measure of spread for the wavefront. The
 751 sample standard deviation was used since this one example does not represent the whole population

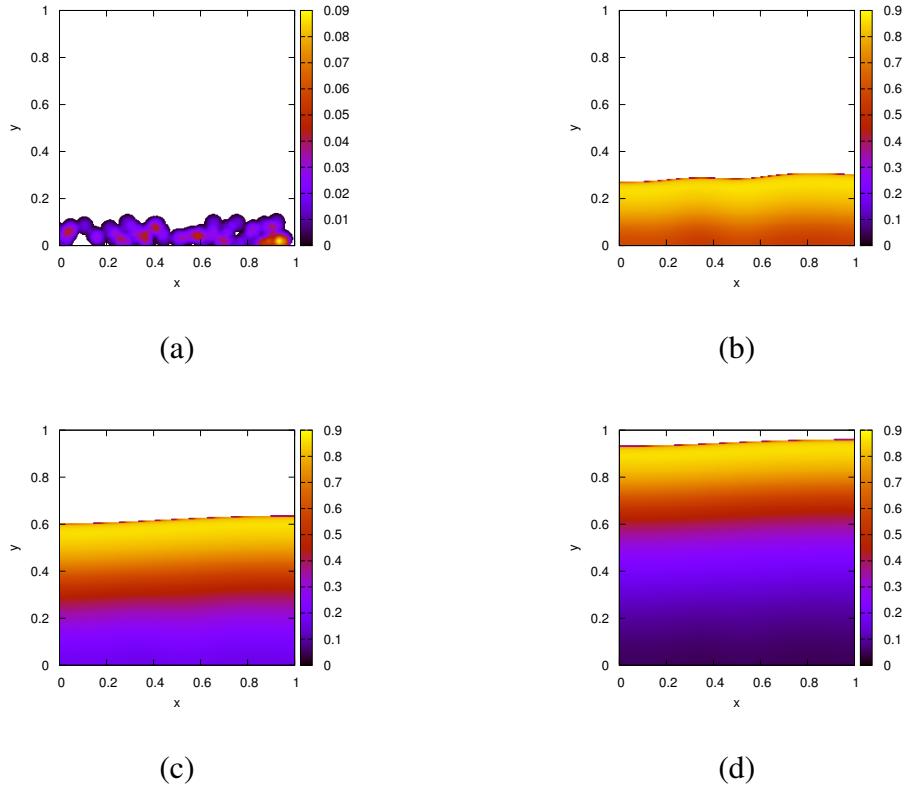


Figure 4.10: Plots of the simulation with random spherical innoculation points centered in the region $(x, y) \in [0, 0] \times [1, 0.1]$. The solutions are shown at (a) $t = 0$, (b) $t = 10$, (c) $t = 20$, and (d) $t = 30$. Each solution is computed on a 513×513 grid.

752 of solutions. The idea is that, for a solution that converges to one-dimensionality, the y location of
 753 the wavefront should be converging to similar values. This means that the standard deviation would
 754 converge to zero. The standard deviation of the wavefront as a function of time of the simulation
 755 ran in Figure 4.10 can be seen in Figure 4.11. Here is shows that the solution is converging to zero,
 756 however not monotonically.

757 For the numerical computation of the wavefront, the largest y values greater than 0.001 was used
 758 instead of 0. The reason is that there are very small values of around 10^{-200} that arise due to the
 759 diffusion that were not adequate representations of the wavefront.

760 Another interesting item to investigate is the actual shape of the wavefront. Figure 4.12 shows only
 761 the wavefront shape for multiple timesteps. The wavefront shape is the same dataset of points used to
 762 calculate the standard deviation of the wavefront interface. Of interest is that the wavefront seems to

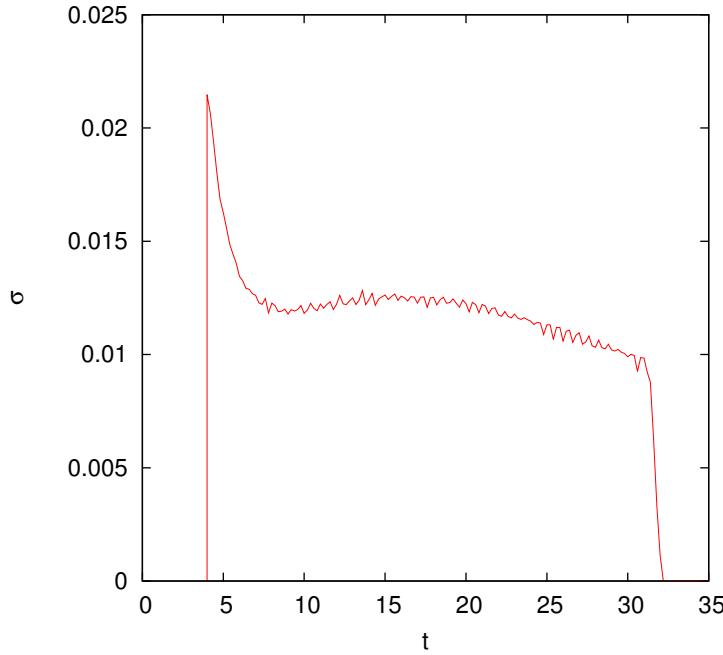


Figure 4.11: The standard deviation of the wavefront interface as a function of time. The wavefront references the largest y coordinate with $M > 0.001$ for each x coordinate. The choice of using $M > 0.001$ is because we want to ignore the small values (10^{-100}) that arise from the diffusion right at the wave front. This simulation is the same as the previous Figure.

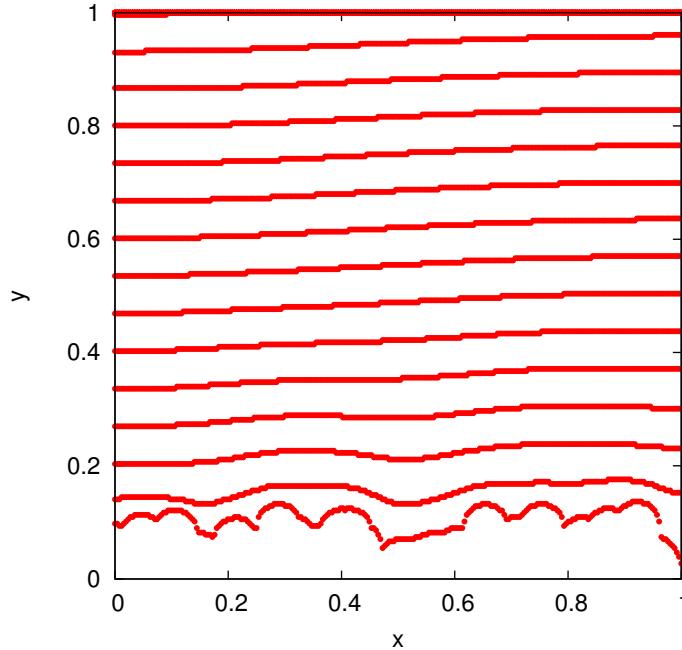


Figure 4.12: The wavefront shape of multiple timesteps. Each wavefront has a difference in time by 2, i.e. they are at $t = 4, 6, 8, \dots, 58, 60$. The simulation results are the same as the previous Figures, using the default parameter values with a grid size 513×513 .

763 move at a constant speed, since each wavefront shown is equidistance from the next.

764 4.2.4 Parameter Effect on Wavespeed

765 The travelling wave solutions seen before have all existed for a single set of parameters. Here the four
 766 main system parameters, δ , κ , ν , and γ are independent varied and the effect on the travelling wave
 767 solutions are observed. From this we can also see how the wavespeed of the travelling wave solution
 768 changes as a function of the different model parameters.

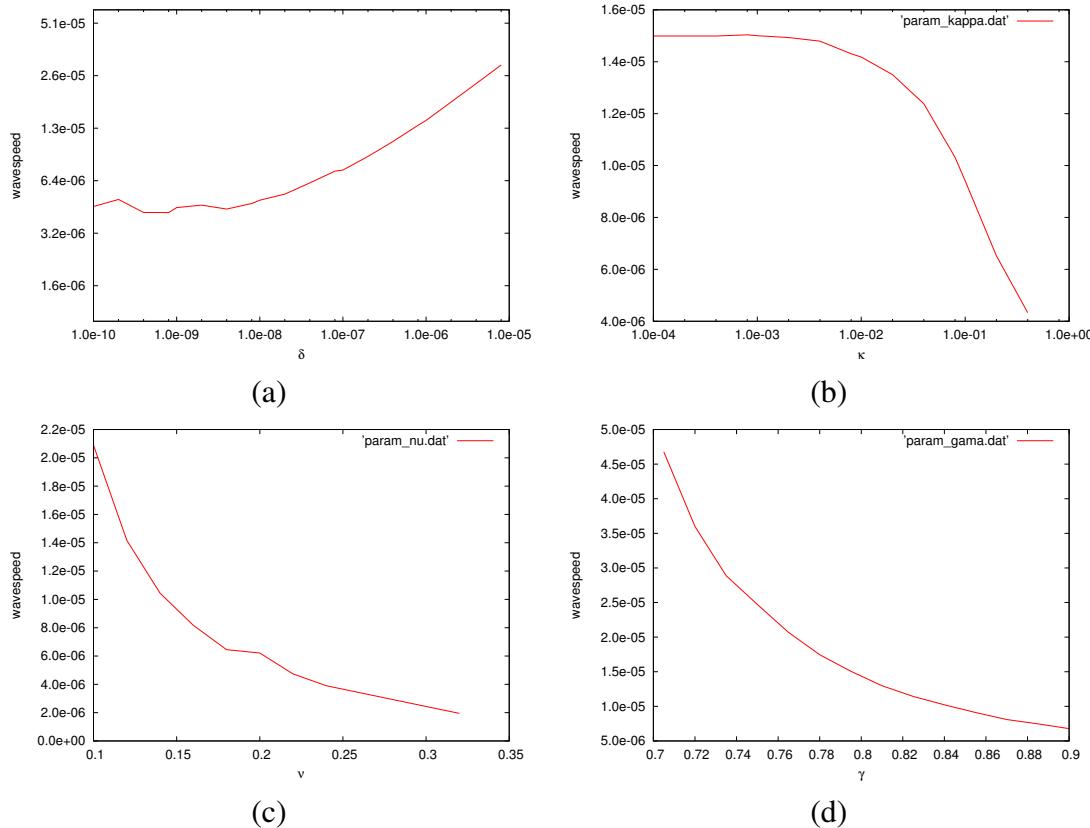


Figure 4.13: The value of c as parameter (a) δ , (b) κ , (c) ν , and (d) γ are changed. Note that (a) and (b) have logscales due to the selection of parameter values. Each of these were calculated with the same setup as the travelling solution previously done. The grid size for each was 513×4 and a time step of $\Delta t = 0.001$ was used.

769 For this an automated script was created that checks, for each timestep, if M is travelling wave solu-
 770 tion based on the solution of M from a number of timesteps previous. From this check, a wavespeed
 771 needs to be approximated based on the distance between the two solutions. If this approximated
 772 wavespeed is matched throughout all the x values, then a travelling wave solution is assumed to exist
 773 at that timestep. With this script, we can try the same simulation as Figure 4.7 with different parameter

774 values.

775 For each parameter, δ , κ , ν , γ , the range of values choosen were arbitrarily. Figure 4.13 shows the
776 results of the wavespeed for each parameter changes. Mainly it was so that the solution did not
777 progegate too fast and hit the end of the region before developing into a full travelling wave solution.
778 Generally when the travelling wave does not form it is because the wave front progegates to the end
779 of the region faster then the tail of the travelling wave can decrease to 0. In the case of ν , any larger
780 values then the selected range resulted in biomass that died faster then it could grow, and thus no
781 travelling wave solution exists. There did not appear to be any cases were a travelling wave solution
782 could not form.

783 Since, for each of the choosen parameters, a travelling wave solution was seen to exist, it can be said
784 that the choice of parameter in Figure 4.7 was not the only reason for the travellin wave solutions
785 existance.

786 4.3 Spatial Effects

787 There are many spatial effect that can exist here because of the diffusion term in the biomass. We
788 now try observing any differences in the behaviour of the system based on spatially different initial
789 conditions. This will show if there is any noticeable differences in the behaviour of the system based
790 solely on the placement of initial biomass. There will be two methods to compare the different
791 simulations: visually and by monitoring the amount of CO_2 produced. The visual inspection is a
792 logical comparison to use, the CO_2 is chosen since it essentially provides a measure of the activity in
793 the system. The CO_2 is also used in the experiments conducted in Dumitrache (2014). Looking at the
794 CO_2 production, a lumped measurement of the biomass activity, shows whether local spatial effects
795 change the global reactor scale behaviour.

796 To measure the difference, two simulations will be run with varying initial conditions. One simulation
797 will have the initial condition evenly spread along one side of the region. The other will have all
798 the initial condition clumped in single corner. This will replicate the two possible extremes for the

799 location of biomass.

800 Since the only difference is the initial conditions, and since an equal start must be assured, the selec-
801 tion of initial condition is important. The equally dispersed initial condition needs to have as much
802 surface area exposed as reasonably possible. To this end, *num* spherical inoculation points, equidis-
803 tance from each other, are used along the $y = 0$ side of Ω . Here we use (x_e, y_e) as the center of the
804 evenly distributed inoculation points. The value of (x_e, y_e) depends on the number of points chosen
805 for the simulation. Each spherical inoculation point is computed as,

$$\text{806} \quad M(0, x, y) = \frac{-h}{d^2}((x - x_e)^2 + (y - y_e)^2) + h, \quad M \geq 0. \quad (4.8)$$

807 Note that $M(0, x, y)$ is for points within circles of radius d centered at (x_e, y_e) , otherwise $M(0, x, y) =$
808 0. For the substratum we have $C(0, x, y) = 1$ everywhere.

809 For the clumped initial condition, we choose another spherical inoculation point so that it is similar
810 to the evenly distributed initial condition. Here, we need only a single inoculation point, centered at
811 the corner $(0, 0)$. The choice of inoculation center is so that the initial biomass is as concentrated as
812 possible, while still retaining a spherical shape. The initial condition for the clumped biomass is as
813 follows,

$$\text{814} \quad M = \frac{-1}{(2hd^2 \cdot num)^{\frac{1}{3}}} (x^2 + y^2) + (2hd^2 \cdot num)^{\frac{1}{3}}. \quad (4.9)$$

815 Here the coefficients have been chosen so that the two initial conditions have the same amount of
816 biomass. The value of *num* is to represent the number of inoculation points in the evenly distributed
817 initial condition.

818 Figure 4.14-4.15 shows the time evolution of both initial conditions. Here we arbitrarily select *num* =
819 6.

820 It becomes easier to see the difference between the two spatially difference problems when CO_2
821 production is taken into account. In Figure 4.16 it is clear that there is a substantial difference between
822 the CO_2 production of both cases. This shows that the initial distribution of biomass can affect a

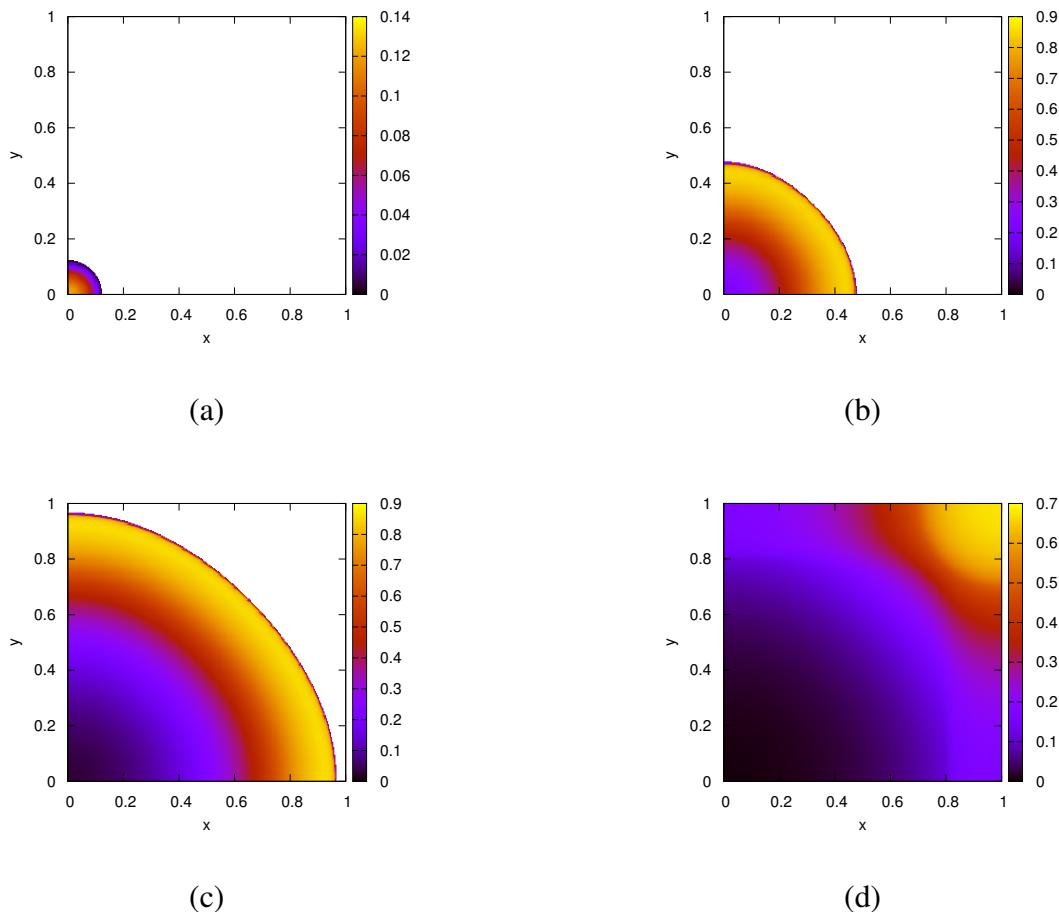


Figure 4.14: This shows the time evolution for the clumped initial condition at (a) $t = 0$, (b) $t = 16$, (c) $t = 32$, (d) $t = 48$. Here default parameters are used on a grid size of 513×513 .

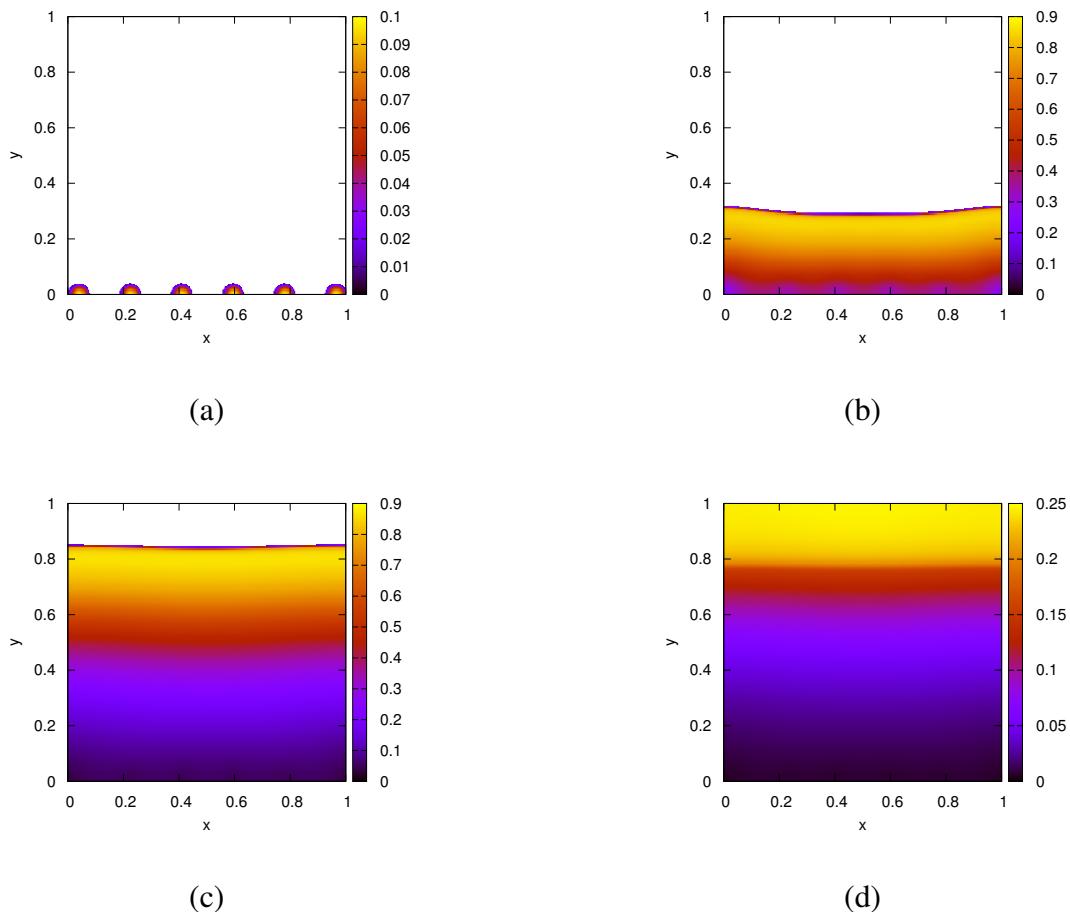


Figure 4.15: This shows the time evolution for the evenly distributed initial condition at (a) $t = 0$, (b) $t = 16$, (c) $t = 32$, (d) $t = 48$. Here default parameters are used on a grid size of 513×513 .

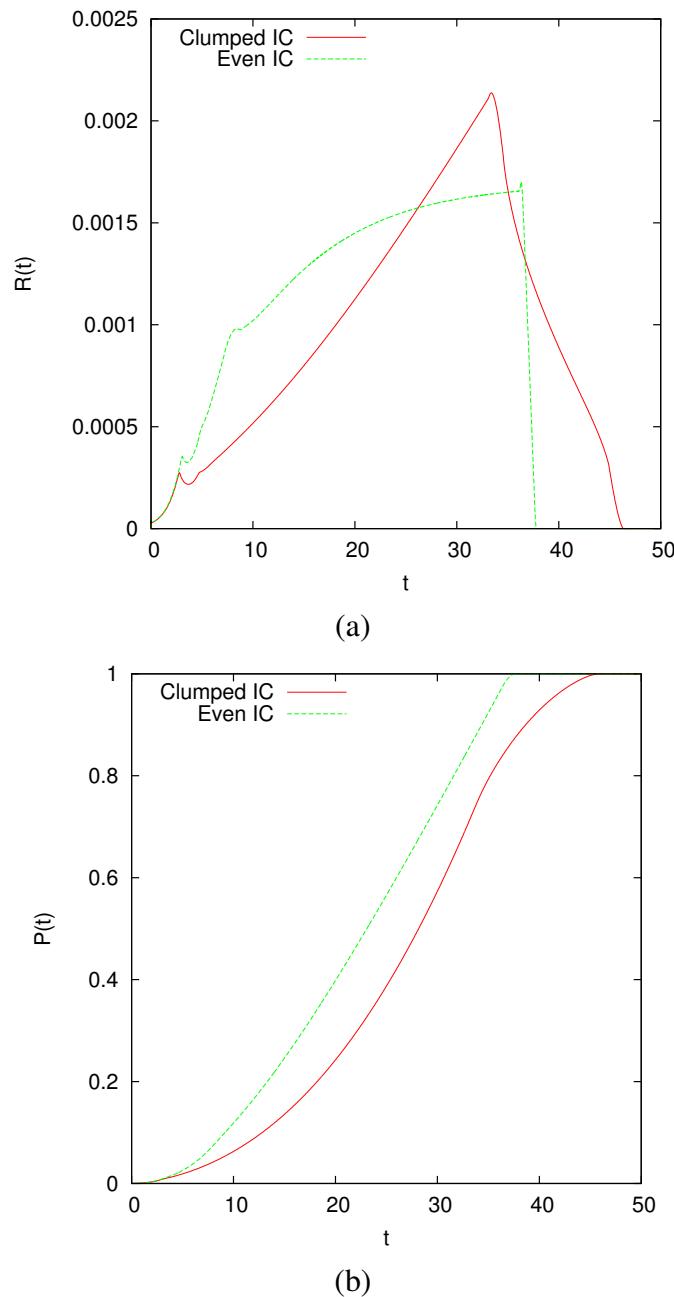


Figure 4.16: A plot of (a) $\mathcal{R}(t)$ and (b) $\mathcal{P}(t)$. These are extracted from the same simulation done in the two previous figures.

823 lumped, global measurement and change the reactor-scale behaviour. From this, two dimensional
824 models that take spatial consideration at a meso-scale can lead to different results.

825 **Chapter 5**

826 **Conclusions**

827 **5.1 Lessons Learned**

828 From completing each of the main objectives, some meaningful lessons can be taken from the results:

- 829 • From the numerics chapter of the thesis, the validity and usefulness of a newly developed
830 fully-implicit method was investigated. By comparing it to the standard semi-implicit method
831 for which it is an extension of it was determined that a significant accuracy gain results from a
832 single extra iteration of the fully-implicit method. Multiple iterations increase this gain since the
833 increase in accuracy is positively correlated to the number of iterations performed. However, the
834 computational effort required from the fully-implicit method grows exponentially with higher
835 tolerance. The best ratio for solution accuracy when weighed against heavier computation
836 times suggests that two iterations of the fully-implicit method is best (one extra from the semi-
837 implicit method). This resulted in an extra digit of accuracy at the cost of approximately 150%
838 the computational effort.
- 839 • From the simulation chapter of the thesis, a number of useful characteristics were observed
840 in the system. The existence of travelling wave solutions was strongly suggested from all the
841 evidence gathered. The stability of this wave suggests that it always exists, but this cannot be
842 verified due to the analytic complexity of the problem. Testing two spatially different simula-

843 tions and measuring the CO_2 production showed a large difference between the two solutions at
844 a reactor-scale. This suggests that two dimensional models are better for accurately mimicing
845 the behaviour of the system.

846 5.2 Future Work

847 There are a number of extensions that could be made of the work done here:

- 848 • What could have been changed/improved/explored/avoided? How could the change be made?
849 What could this gain? What possible challenges could this change make?

850 **References**

- 851 Barrett, R., Berry, M., T.F., C., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine,
852 C., and van der Vorst, H. (1987). *Templates for the Solution of Linear Systems: Building Blocks for*
853 *Iterative Methods*. Society for Industrial and Applied Mathematics, 1st edition.
- 854 Burden, R. and Faires, J. (2010). *Numerical Analysis*. Brooks/Cole, 9th edition.
- 855 Butcher, J. (2008). *Numerical Methods for Ordinary Differential Equations*. Wiley, 2nd edition.
- 856 Duddu, R., Bordas, S., Chopp, D., and Moran, B. (2008). A combined extended finite element and
857 level set method for biofilm growth. *International Journal for Numerical Methods in Engineering*,
858 74:848–870.
- 859 Dumitrache, A. (2014). *Understanding Biofilms of Anaerobic, Thermophilic and Cellulolytic Bacteria: A Study towards the Advancement of Consolidated Bioprocessing Strategies*. PhD thesis,
860 University of Toronto.
- 861 Dumitrache, A., Eberl, H., Wolfaardt, G., and Allen, D. (2015). Mathematical modeling to validate
862 on-line CO_2 measurements as a metric for cellulolytic biofilm activity in continuous-flow bioreac-
863 tors. *Biochemical Engineering Journal*, 101:55–67.
- 864 Dumitrache, A., Wolfaardt, G., Allen, D., Liss, S., and Lynd, L. (2013a). Form and function of
865 clostridium thermocellum biofilms. *Applied and Environmental Microbiology*, 79:231–239.
- 866 Dumitrache, A., Wolfaardt, G., Allen, D., Liss, S., and Lynd, L. (2013b). Tracking the cellulolytic
867 activity of clostridium thermocellum biofilms. *Biotechnology for Biofuels*, 6:175.

- 869 Eberl, H. and Demaret, L. (2007). A finite difference scheme for a doubly degenerate diffusionreaction
870 equation arising in microbial ecology. *Electronic Journal of Differential Equations*, CS15:77–
871 95.
- 872 Eberl, H., Khassehkhan, H., and Demaret, L. (2010). A mixed-culture model of a probiotic biofilm
873 control system. *Computation and Mathematical Methods in Medicine*, 11:99–118.
- 874 Eberl, H., Parker, D., and van Loosdrecht, M. (2001). A new deterministic spatio-temporal continuum
875 model for biofilm development. *Journal of Theoretical Medicine*, 3:161–175.
- 876 Efendiev, M., Eberl, H., and Zelik, S. (2002). Existence and longtime behaviour of solutions of a
877 nonlinear reaction-diffusion system arising in the modeling of biofilms. *RIMS Kokyuroko*, 1258:49–
878 71.
- 879 Gallo, C. and Manzini, G. (2003). Finite volume/mixed finite element analysis of pollutant trans-
880 port and bioremediation in heterogenous saturated aquifers. *International Journal for Numerical
881 Methods in Fluids*, 42:1–21.
- 882 Gurtin, M.E., M. (1977). On the diffusion of biological populations. *Mathematical Biosciences*,
883 33:35–49.
- 884 Jalbert, E. and Eberl, H. (2014). Numerical computation of sharp travelling waves of a degenerate
885 diffusion-reaction equation arising in biofilm modelling. *Communications in Nonlinear Science
886 and Numerical Simulation*, 19(7):2181–2190.
- 887 Khassehkhan, H. and Eberl, H. (2008). Modeling and simulation of a bacterial biofilm that is con-
888 trolled by ph and protonated lactic acids. *Computation and Mathematical Methods in Medicine*,
889 9(1):47–67.
- 890 Khassehkhan, H., Hillen, T., and Eberl, H. (2009). A nonlinear master equation for a degenerate
891 diffusion model of biofilm growth. *Lecture Notes in Computer Science*, 5544:735–744.

- 892 Macías-Díaz, J., Macías, S., and Medina-Ramírez, I. (2013). An efficient nonlinear finite-difference
893 approach in the computational modeling of the dynamics of a nonlinear diffusion-reaction equation
894 in microbial ecology. *Computational Biology and Chemistry*, 47:24–30.
- 895 McCollum, J., Peterson, G., Cox, C., Simpson, M., and Samatova, N. (2006). The sorting direct
896 method for stochastic simulation of biochemical systems with varying reaction execution behavior.
897 *Computational Biology and Chemistry*, 30(1):39–49.
- 898 Noguera, D., Pizarro, G., Stahl, D., and Rittmann, B. (1999). Simulation of multispecies biofilm
899 development in three dimensions. *Water Science and Technology*, 39:123–130.
- 900 Picioreanu, C., van Loosdrecht, M., and Heijnen, J. (1998). A new combined differential-discrete cel-
901 lular automaton approach for biofilm modeling: application for growth in gel beads. *Biotechnology*
902 and *Bioengineering*, 57:718–731.
- 903 Rittmann, B. and McCarty, P. (1980). Model of steady-state biofilm kinetics. *Biotechnology and*
904 *Bioengineering*, 22:2343–2357.
- 905 Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied
906 Mathematic, 2nd edition.
- 907 Sirca, S. and Horvat, M. (2012). *Computational Methods for Physicistsl: Compendium for Students*.
908 Springer.
- 909 Varga, R. (2004). *Geršgorin and His Circles*. Berlin: Springer-Verlag.
- 910 Wang, Z.-W., Lee, S.-H., Elkins, J., and Morrell-Falvey, J. (2011). Spatial and temporal dynamics
911 of cellulose degradation and biofilm formation by caldicellulosiruptor obsidiansis and clostridium
912 thermocellum. *AMB Express*, 1:30.
- 913 Wanner, O., Eberl, H., Morgenroth, E., Noguera, D., Picioreanu, C., Rittmann, B., and van Loos-
914 drecht, M. (2005). *Mathematical modeling of biofilms*. IWA Scientific and Technical Report no
915 18., UK: IWA Publishing.

916 **Appendix A**

917 **Default Parameter Values**

918 The default parameter values used for simulation are listed in the follow table. Unless stated, every
919 simulation uses these values.

Parameter	Symbol	Value
-	α	4
-	β	4
Death rate	ν	0.12
Half-concentration rate	κ	10^{-3}
Yield rate	γ	0.80
Diffusion constant	δ	10^{-4}
Initial condition height	h	0.1
Initial condition depth	d	$\frac{5}{127}$
Number of grid points	nm	513×513
Grid size	Δx	$\frac{1}{513}$
Time step	Δt	10^{-3}

Table A.1: The listing of default parameter values used for most simulations.