Comparing Semi-Implicit and Full-Implicit Method for Solving Stiff Density Dependent Diffusion-Reaction Equations Arising in Biofilm Growth Models

Hermann J. Eberl and Eric M. Jalbert
(heberl@uoguelph.ca, ejalbert@uoguelph.ca)
Dept. of Mathematics and Statistics, University of Guelph,
Guelph, ON N1G 2W1, Canada

Abstract. Using the previously established mathematical model for the growth of *Clostridium thermocellum* as an example, a comparison of a semi-implicit and fully-implicit numerical method will be conducted. This example is chosen specifically because it is a stiff problem. The semi-implicit method follows the idea of the semi-implicit Euler method. The fully-implicit method is similar except it computes the solutions multiple times for each time step, each using the newly computed values inplace of the solutions at the next time step. These two methods show a minute difference in accuracy, the fully-implicit method being mildly more correct. The difference in computational intensity is more pronouced, the semi-implicit method typically between 3 and 4 times faster. The two values compare nicely and, given the correct situation, both have their uses. Ultimately, the gain in accuracy from the fully-impliciti method does generally justify the increase in accuracy; showing the semi-implici method as a sufficient numerical method for these types of problems.

1 Intro

Bacterial biofilms are communities of microorganisms that adhere on immersed aquaeous surfaces that can support microbial growth based on the environmental conditions. Biofilms are prevalent on both organic or inorganic surfaces in natural, industrial, or hospital settings. When attached to a surface, these bacterias embed themselves in a self-produced extracellular polymeric substance (EPS). The EPS provides a layer of protection against washout and biocides, making harmful biofilms difficult to remove. This attribute of biofilms is a problem since they are associated with medical issues in the form of, bacterial infections, dental plaque, and other diseases and industrial issues, such as biocorrosion of water pipes. They also contribute positivly to wastewater treatment, groundwater protection, soil remediation, and other environmental engineering technologies.

To further the understanding of biofilms, a diffusion-reaction model for biofilm growth was proposed in [4]. This model is based on this work.

Through the use of some of many established numerical method, an analysis of this biological system can be completed. Typically, in these kind of time-dependent systems, one would discritize both space and time to facilitate obtaining the solution. The numerical methods can be classified as explicit, semi-implicit, or fully-implicit based on the methods dependence for the state of the system. An explicit numerical method uses the current state of the system to

compute the next state while a fully-implicit is the opposite [2]. A semi-implicit method is specific to a system of differential equations, where at least one equation is solved using the current state and another is solved using the next state. This difference between semi-implicit and fully-implicit numerical methods is that using a fully-implicit method is more computationally heavy, but gives greater accuracy.

2 Model

The specific biology that is discussed here is the growth of *Clostridium thermocellum* biofilms on cellulose sheets. This cellulolytic anaerobic bacteria is a special type of biofilm, it displays uncharacteristic behaviour by not generating a EPS. Based on the work of [3], a simple mathematical model to simulate the growth of *C. thermocellum* can be created. They modelled the biomass density and nutrient concentration as a two PDE-coupled system. Here the spatial diffusion of the nutrient concetration is removed to mimic the carbon substrait that *C.Thermocellum* consumes to grow. The PDE-ODE coupled system is purposed as,

$$M_t = \nabla_x \left(D(M)\nabla_x M\right) + f(C, M)M \tag{1}$$

$$C_t = -g(C)M \tag{2}$$

where

$$D(M) = \delta \frac{M^\alpha}{(1-M)^\beta} \tag{3}$$

$$f(C, M) = g(C) - \frac{y}{10} \tag{4}$$

$$g(C) = \frac{yC}{k + C}. \tag{5}$$

The spatial diffusion of biomass M is a density-dependent diffusion, given by (3). Here, δ depicts the diffusion rate of the biomass. The growth rate of biomass, $f(C, M)$, is given by (4). The growth rate is Monod kinetic growth with a death term. There exists a yield term, y, that controls the amount of nutrients the biomass gains from the substraits. In (2) there is only a consumption term from the bacteria consuming the carbon substrait. Emphasis will be made to point out that (2) does not have spatial dependencies, since the cellulose sheet is a solid fiber that cannot move.

2.1 Nondimensionalization

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

$$x = \frac{\chi}{L} \implies Ldx = d\chi \tag{6}$$

$$t = \mu\tau \implies \frac{1}{\mu}dt = d\tau \tag{7}$$

$$C = \frac{\overline{C}}{C_0} \tag{8}$$

$$\delta = \frac{1}{\mu L^2}\overline{\delta} \tag{9}$$

$$k = \frac{\overline{k}}{C_0} \tag{10}$$

$$\nu = \frac{\overline{\nu}}{\mu C_0} \tag{11}$$

$$\mu = \overline{\mu} \tag{12}$$

This gives the system

$$M_t = \frac{1}{\mu L^2}\nabla_x\left(\hat{D}(m)\nabla_x M\right) + \frac{1}{\mu}\hat{f}(C,M) \tag{13}$$

$$C_t = \frac{-1}{C_0\mu}\hat{g}(C,M) \tag{14}$$

where

$$\hat{D}(M) = \mu L^2\delta\frac{M^\alpha}{(1-M)^\beta} \tag{15}$$

$$\hat{f}(C,M) = \frac{\mu C C_0}{kC_0 + CC_0}M\left(1 - \left(\frac{M}{C}\frac{M_0}{C_0}\right)^\gamma\right) \tag{16}$$

$$\hat{g}(C,M) = -\frac{\mu C_0 \nu C C_0}{kC_0 + CC_0}M \tag{17}$$

This can be greatly simplified by cancelling out parameters.

$$M_t = \nabla_x\left(\delta\frac{M^\alpha}{(1-M)^\beta}\nabla_x M\right) + \frac{C}{k+C}M\left(1 - \left(\frac{M}{C}\frac{M_0}{C_0}\right)^\gamma\right) \tag{18}$$

$$C_t = -\frac{\nu C}{k+C}M \tag{19}$$

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

$$M_t = \nabla_x\left(D(M)\nabla_x M\right) + f(C,M) \tag{20}$$

$$C_t = -g(C,M) \tag{21}$$

where

$$
\begin{aligned}
D(M) &= \delta \frac{M^{\alpha}}{(1-M)^{\beta}} \\
f(C,M) &= \frac{C}{k+C} M \left(1 - \left(\frac{M}{C}\frac{M_0}{C_0}\right)^{\gamma}\right) \\
g(C,M) &= \frac{\nu C}{k+C} M
\end{aligned}
\tag{22}
$$

This system is solved on a non-dimensionalized square region $\Omega \in [0,1] \times [0,1]$. On this region, there are Neumann boundary conditions defined as

$$
\frac{\partial M}{\partial x} = \frac{\partial C}{\partial x} = 0, \quad \text{on } \partial\Omega. \tag{23}
$$

3 Method

There are three main aspects to the numerical method of solving this system. The system must be temporally and spatially discritized, then a solution for C and M can be computed using numerical methods.

For the solution of C, since it has no spatial diffusion, a simple application of the Trapezidral rule can be used. The idea is to integrate both sides of (2) to get,

$$
C^{t_{n+1}} - C^{t_n} = \frac{t_{n+1} - t_n}{2} \left(g(C^{t_{n+1}}, M^{t_{n+1}}) + g(C^{t_n}, M^{t_n})\right). \tag{24}
$$

By rearranging (24) and subbing in (5) an equation explicitly for $C_{t_{n+1}}$ is found. This solution gives two values for $C_{t_{n+1}}$ since it is solved using the quadratic formula. Only the positive branch of the quadratic formula is relavent to this problem, since the negative branch behaves unrealisticly.

The solution of M is more involved, because of the density-dependent diffusion term. This means that it must be spatially discritized as well as temporally discritized. The spatial discritization of the system is a simple application of the finite difference method. Following the example of finite differences for 2-D problems as described in [6] a grid is created to approximate each point using a five-point stencil. Here, the values at the boundaries are unknown and as such are also included in the five-point stencil These boundary points are computed using a second order appoximation. The temporal discritization is a forward difference between the current and next time step. Applying these discritizations

to (1) and rearranging gives a system of linear equation defined by,

$$\begin{aligned}\frac{M^n_{i,j}}{\Delta t} &= \frac{-D^n_{i,j-\frac{1}{2}}}{\Delta y^2}\cdot M^{n+1}_{i,j-1} + \frac{-D^n_{i-\frac{1}{2},j}}{\Delta x^2}\cdot M^{n+1}_{i-1,j} \\ &+ \left[\frac{D^n_{i,j-\frac{1}{2}}}{\Delta y^2} + \frac{D^n_{i-\frac{1}{2},j}}{\Delta x^2} + \frac{D^n_{i+\frac{1}{2},j}}{\Delta x^2} + \frac{D^n_{i,j+\frac{1}{2}}}{\Delta y^2} - f(C^n, M^n) + \frac{1}{\Delta t}\right]\cdot M^{n+1}_{i,j} \\ &+ \frac{-D^n_{i+\frac{1}{2},j}}{\Delta x^2}\cdot M^{n+1}_{i+1,j} + \frac{-D^n_{i,j+\frac{1}{2}}}{\Delta y^2}\cdot M^{n+1}_{i,j+1},\end{aligned} \tag{25}$$

where $M^n_{i,j} = M^{t_n}(x_i, y_j)$, $D^n_{i,j} = D(M^n_{i,j})$, and (x_i, y_j) is the ordered pair at grid point i, j.

Using (25) a five-diagonal block matrix can be created, defined as,

$$A = \begin{pmatrix} M_{i,j} & M_{i+1,j} & & M_{i,j+1} & & & & & & \\ M_{i-1,j} & \ddots & \ddots & & \ddots & & & & & \\ & \ddots & \ddots & \ddots & & \ddots & & & & \\ M_{i,j-1} & & M_{i-1,j} & M_{i,j} & M_{i+1,j} & & M_{i,j+1} & & & \\ & \ddots & & \ddots & \ddots & \ddots & & \ddots & & \\ & & M_{i,j-1} & & M_{i-1,j} & M_{i,j} & M_{i+1,j} & & M_{i,j+1} & \\ & & & \ddots & & \ddots & \ddots & \ddots & & \\ & & & & \ddots & & \ddots & \ddots & M_{i+1,j} & \\ & & & & & M_{i,j-1} & & M_{i-1,j} & M_{i,j} & \end{pmatrix}, \tag{26}$$

where each $M_{i,j}$ is the coefficient based on (25).

The solving of (26) can be completed through use of a linear solver. According to [1], if (26) is positive definite and symmetric then it can be solved using the Conjugate Gradiant method.

Proposition 1. *The matrix A, defined in (26), is positive definite and symmetric.*

Proof. Matrix A is positive definite if all the eigenvalues are positive. Using Geršgorin's circle theorem, see [5], the eigenvalues can be shown to be positive if, on all rows independently, the sum of the off-diagonals values are less then the diagonal value. Mathematically we have,

$$\begin{aligned}&\left|\frac{-D^n_{i,j-\frac{1}{2}}}{\Delta y^2} + \frac{-D^n_{i-\frac{1}{2},j}}{\Delta x^2} + \frac{-D^n_{i+\frac{1}{2},j}}{\Delta x^2} + \frac{-D^n_{i,j+\frac{1}{2}}}{\Delta y^2}\right| \\ &\quad < \left|\frac{D^n_{i,j-\frac{1}{2}}}{\Delta y^2} + \frac{D^n_{i-\frac{1}{2},j}}{\Delta x^2} + \frac{D^n_{i+\frac{1}{2},j}}{\Delta x^2} + \frac{D^n_{i,j+\frac{1}{2}}}{\Delta y^2} - f(C^n, M^n) + \frac{1}{\Delta t}\right|,\end{aligned} \tag{27}$$

which simplifies to,

$$f(C^n_{i,j}, M^n_{i,j}) < \frac{1}{\Delta t}, \tag{28}$$

which is true give the values of Δt that are used. Therefore we have that A is positive definite.

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

Given that A is positive definite and symmetric, the conjugate gradiant method can be used to compute the solution. As an added property, A also happens to be diagonally dominate. This means that it could be solved using Bi-Conjugate Gradient Method. However the Conjugate Gradient method has a faster computation time then Bi-Conjugate Gradiant method for this problem and is used for this reason [1].

Now that the method for solving C and M have been discussed, the method for solving the system as a whole can be investigated. Since $M^{t_{n+1}}$ only depends on C^{t_n} and M^{t_n} while $C^{t_{n+1}}$ depends on C^{t_n}, M^{t_n}, and $M^{t_{n+1}}$ the solution for M should be computed first and then it can be used to compute C. This idea is translated into Algorithm 1 to be the fully-implicit method for solving the system. The semi-implicit method for solving the system is the same as Algorithm 1 but with $eSoln = 0$, forcing only a single iteration of the loop.

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

Algorithm 1: Algorithm for the fully-implicit solving of (1-2).

4 Results

4.1 Simulation Setup

All the simulations were run on a custom built workstation with an Intel Xeon CPU E5-2650 (1.2 GHz, 20 MB cache size) and 32 GB RAM under Red Hat Enterprise Linux Server release 6.5 (Santiago). Running the simulations with OpenMP, took advantage of the 16 processers of the Intel Xeon CPU, each with 2 threads. The GNU fortran compiler version 4.4.7 was used for the typical simulation results; compiler arguments were

```
-O3 -fdefault-real-8 -fopenmp .
```

The Intel Fortran compiler version 14.0.2 was used for the speed results; compiler arguments were

```
-O3 -r8 -openmp.
```

The simulations were solved on a 1024×1024 grid, using $\Delta t = 10^{-2}$.

4.2 Results

Using the parameter values defined in the Appendix, the simultation was run and the results were as shown in Figure 1.

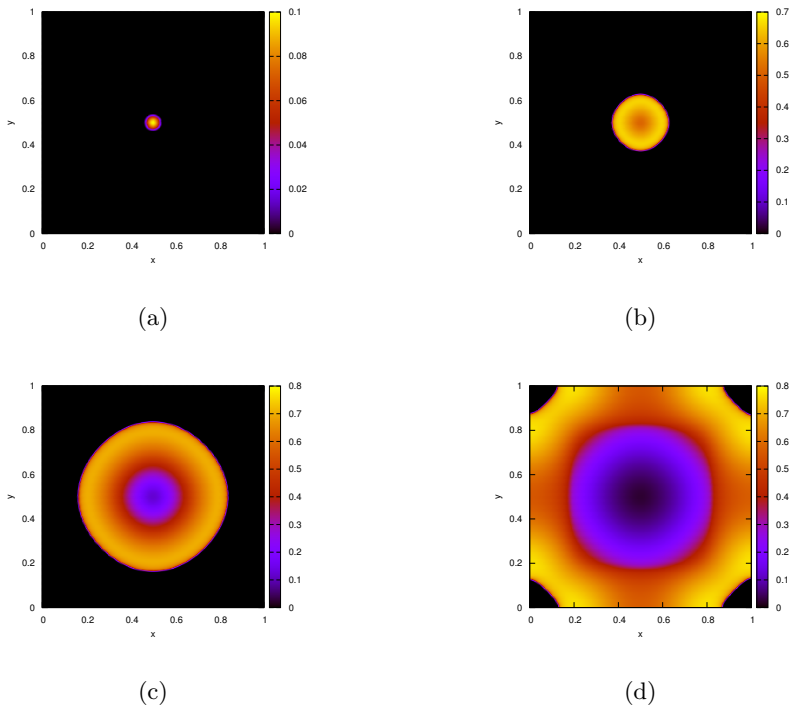

Fig. 1. Typical simulation using parameters from Appendix at (a) $t = 0$, (b) $t = 10$, (c) $t = 20$, (d) $t = 30$

4.3 Comparisons

A comparison of the the two methods can be done by looking at both their computation time and the accuracy of the solution. The computation time is computed by the system clock from the start of the algorithm to the end of the algorithm. The accuracy of the solution can be found by computing the normed-difference between two solutions, as follows,

$$\epsilon_{sol} = \frac{||u_1 - u_2||}{||u_1||} \tag{29}$$

Two comparisons will be done with the norm. One with the solution at the current grid size and the solution at the next smallest grid size. This is because, the samller the grid refinement, the more accurate the solution. Thus to compare the accuracy of methods, this will be used. The other will be with the solution and the solution with the smallest $eSoln$, as defined in Algorithm 1. These two comparison will provide two difference measures of the accuracy of the method.

The other comparison will be with where the method is spending its time. The fully-implicit method should have less iterations of the linear solver. Thus with will be measured as $Iteration2$. $Iterations1$ is the number of iterations between solving M and C, according to Algorithm 1.

The results of the method comparison can be seen in Tabel 1.

Tol. 1	Computation Time	ϵ_{sol}	Avg. Iter. 1	Max Iter. 1	Avg. Iter. 2	Max Iter. 2
10^0	1430.05	—-	1	1	44.08	72
10^{-2}	—-	—-	—-	—-	—-	—-
10^{-4}	—-	—-	—-	—-	—-	—-
10^{-8}	—-	—-	—-	—-	—-	—-

Table 1. Results from running simulation with different Tolerance 1 values. Note, Tolerance 1 of 10^0 is the semi-implicit method.

5 Conclusions

The numerical simultating of this problem can be solved in many differenct manners. Using the fully-implicit method as described in Algorithm 1 allows for an extra degree of accuracy for the solution. That being said, the difference in computation time is not neccessarily worth the increased workload. The semi-implicit method has shown to provide solution that are sufficiently accurate and thus the extra computations for the fully-implicit method are not always needed. The choice of method is mostly dependent on the problem that is being solved and the required accuracy of the solution. With powerful computers becoming more readily available, the sacrifice of computation speed becomes a smaller price.

Appendix

Variable/Parameter	Dimensions	Parameter Value
x	$[days]$	$-$
t	$[meters]$	$-$
M	$[-]$	$-$
C	$[\frac{grams}{meters^3}]$	$-$
δ	$[\frac{meters^2}{days}]$	10^{-12}
α	$[-]$	4
β	$[-]$	4
y	$[days^{-1}]$	6
k	$[\frac{grams}{meters^3}]$	4

Table 2. List of parameters and their dimensions

References

1. R. Barrett, M. Berry, Chan T.F., J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Society for Industrial and Applied Mathematics, 1st edition, 1987.
2. J.C. Butcher. *Numerical Methods for Ordinary Differential Equations*. Wiley, 2nd edition, June 2008.
3. A. Dumitrache. *Understanding Biofilms of Anaerobic, Thermophilic and Cellulolytic Bacteria: A Study towards the Advancement of Consolidated Bioprocessing Strategies*. PhD thesis, University of Toronto, 2014.
4. H.J. Eberl, D.F. Parker, and van Loosdrecht M.C.M. A new deterministic spatio-temporal continuum model for biofilm development. *Journal of Theoretical Medicine*, pages 161–175, 2001.
5. Geršgorin S. über die abgrenzung der eigenwerte einer matrix. *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na*, pages 749–754, 1931.
6. Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematic, 2nd edition, 2003.