Simulating Solid Tumor Growth Using Multigrid Algorithms: Mid-Year Report

Asia Wyatt
Applied Mathematics, Statistics, and Scientific Computation Program

Advisor:

Doron Levy
Department of Mathematics/CSCAMM

Abstract

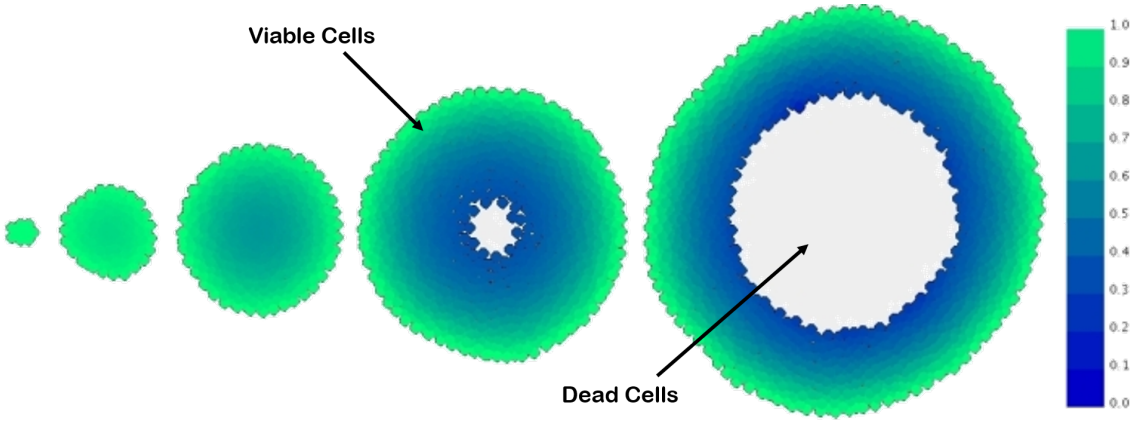
In mathematical cancer modeling, the growth of solid tumors combines the mechanics of cell-cell adhesion, cell growth velocity and angiogenesis amongst numerous others. This project simulates solid tumoral growth with the use of a coupled system of Cahn-Hilliard-type convection-reaction-diffusion equations. This mathematical model uses a two-cellular tumoral structure of viable, or proliferating, cells and dead cells in the necrotic core. The nonlinear system is then discretized in space using a finite difference approximation and in time using a Crank-Nicolson-type algorithm to account for the stiffness in the fourth order diffusion term of the equation. In order to solve the discretized system, a multigrid scheme with uniform square grids is then implemented. Finally, the growth of solid tumors is simulated with varying initial conditions.

The University of Maryland, College Park

1 Motivation

When observing the growth of tumors there are certain constants which come into play. As seen in Figure 1, over time we observe the viable, or proliferating cells on the boundary of the tumor (green), the dormant cells inside of the proliferating cells (blue), and the dead cells within the necrotic core (white). The change in the volume fraction of these growing cells is in direct relationship with the nutrient provided to the tumor, the cellular adhesion, the cellular pressure, and the death rate of the cells to name a few.

Figure 1: The growth of tumors over time [3]



2 Project Goal

The goal of this project is to simulate solid tumor growth in two and, time permitting, three dimensions with a Cahn-Hilliard-type convection-reaction-diffusion mathematical model using multigrid algorithms described in [5].

3 Approach

3.1 Mathematical Model

The mathematical model derived in this section is that which is described in Wise et al. [5]

Coupled System

If ϕ represents the tissue volume fraction, the growth of tumoral tissue reduced down to two cellular—viable and dead cells—is modeled as

$$\phi_V + \phi_D + \phi_H = 1 \tag{1}$$

$$\phi_T = \phi_V + \phi_D \tag{2}$$

$$\frac{\partial \phi_T}{\partial t} = M\nabla(\phi_T \nabla \mu) + S_T - \nabla \cdot (\boldsymbol{u_S}\phi_T) \tag{3}$$

$$\frac{\partial \phi_D}{\partial t} = M\nabla(\phi_T \nabla \mu) + S_D - \nabla \cdot (\boldsymbol{u_S}\phi_D) \tag{4}$$

$$0 = \nabla \cdot (D(\phi_T)\nabla n) + T_c(\phi_T, n) - n(\phi_T - \phi_D) \tag{5}$$

where,

$$\mu = f'(\phi_T) - \varepsilon^2 \nabla^2 \phi_T \tag{6}$$

$$f(\phi) = \phi^2(1-\phi)^2/2 \tag{7}$$

$$\nabla \cdot \boldsymbol{u_S} = S_T \tag{8}$$

$$\boldsymbol{u_S} = -\kappa(\phi_T, \phi_D)(\nabla p - \frac{\gamma}{\varepsilon}\nabla \phi_T) \tag{9}$$

$$S_T = nG(\phi_T)\phi_V - \lambda_L \phi_D, \tag{10}$$

$$S_D = (\lambda_A + \lambda_N \mathscr{H}(n_N - n))(\phi_T - \phi_D) - \lambda_L \phi_D, \tag{11}$$

where M is the mobility constant related to phase separation between tumoral and healthy tissue and S_T and S_D are the net and dead source of tumoral cells, respectively.

When looking strictly at equation (3),

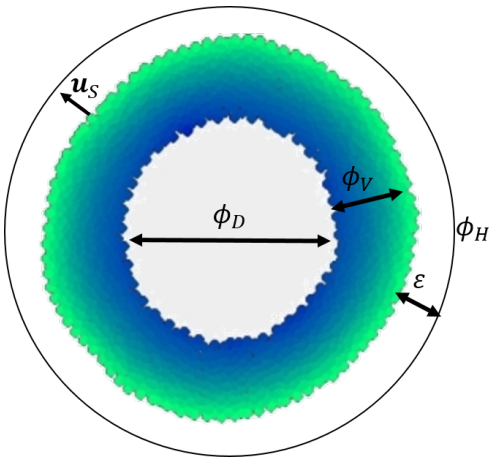
$$\frac{\partial \phi_T}{\partial t} = M\nabla(\phi_T \nabla \mu) + S_T - \nabla \cdot (\boldsymbol{u_S}\phi_T),$$

we have a diffusion term, a source term, and a convection term.

Model Parameters

- ϕ_V: volume fraction of viable tissue
- ϕ_D: volume fraction of dead tissue
- ϕ_H: volume fraction of healthy tissue
- $\boldsymbol{u}_S$: tissue velocity
- ε: thickness of interface between healthy and tumoral tissue
- p: cell-to-cell (solid) pressure
- n: nutrient concentration

Figure 2: Model Parameters on Tumor [3]



Tissue Velocity Equations

The tissue velocity is

$$\boldsymbol{u_S} = -\kappa(\phi_T, \phi_D)(\nabla p - \frac{\gamma}{\varepsilon}\nabla\phi_T) \tag{12}$$

with κ the tissue motility function and $\gamma \geq 0$ the excess adhesion force at the diffuse tumor/host-tissue interface. Assuming the host tissue remains constant we get,

$$\nabla \cdot \boldsymbol{u_S} = S_T. \tag{13}$$

Combining (11) and (12) we have a Poisson equation in p,

$$-\nabla \cdot (\kappa(\phi_T, \phi_D)\nabla p) = S_T - \nabla \cdot (\kappa(\phi_T, \phi_D)\frac{\gamma}{\varepsilon}\nabla\phi_T). \tag{14}$$

Thus, p can be solved for independently then used to find $\boldsymbol{u_S}$ in equation (11).

Source Terms

Assuming that the net source of the tumor cells, S_T, is due to cell proliferation, we have

$$S_T = nG(\phi_T)\phi_V - \lambda_L\phi_D, \tag{15}$$

where λ_L is the rate at which the lysing of dead cells form water and

$$G(x) = \begin{cases} 1 & \text{if } \frac{3\varepsilon}{2} \leq \phi \\ \phi/\varepsilon - \frac{1}{2} & \text{if } \frac{\varepsilon}{2} < \phi < \frac{3\varepsilon}{2} \\ 0 & \text{if } \phi \leq \frac{\varepsilon}{2}. \end{cases} \tag{16}$$

Assuming that the net source of the dead cells, S_D, is due to apoptosis and necrosis with rates λ_A and λ_N, respectively, we have

$$S_D = (\lambda_A + \lambda_N\mathscr{H}(n_N - n))(\phi_T - \phi_D) - \lambda_L\phi_D, \tag{17}$$

where $\mathscr{H}$ is the Heaviside function and n_N is the necrotic limit, which below the tumor tissue dies due to lack of nutrients.

Nutrient Equation

Assuming (i) nutrient diffusion occurs on a faster time scale than cell proliferation and (ii) nutrient taken by dead cells and healthy cells is negligible in comparison to that of viable cells we have,

$$0 = \nabla \cdot (D(\phi_T)\nabla n) + T_c(\phi_T, n) - n(\phi_T - \phi_D), \tag{18}$$

where

$$D(\phi) = D_H(1 - Q(\phi_T)) + Q(\phi_T), \tag{19}$$

$$T_c(\phi_T, n) = (v_p^H(1 - Q(\phi_T)) + v_p^T Q(\phi_T))(n_c - n), \tag{20}$$

and

$$Q(\phi) = \begin{cases} 1 & \text{if } 1 \leq \phi \\ 3\phi^2 - 2\phi^3 \ \textbf{or} \ \phi & \text{if } 0 < \phi < 1 \\ 0 & \text{if } \phi \leq 0. \end{cases} \tag{21}$$

3.2 Scientific Computation Algorithms

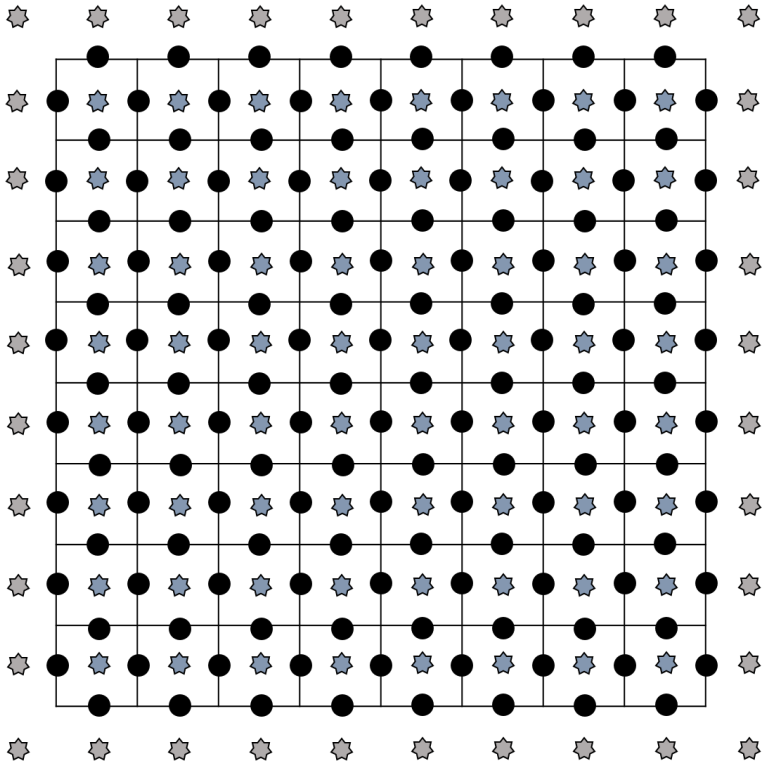
Following the process in Wise et al. [5] the mathematical model for solid tumor growth in two dimensions will be discretized spatially by finite difference and WENO schemes and in time using an implicit Crank-Nicolson-type scheme due to the fourth-order diffusion term. With an explicit scheme, this fourth order term would require a time step restriction of $s < Ch^4$ where s and h are the time and spatial steps, respectively. Though Backward Euler is also an implicit scheme the third-order accuracy of the Crank-Nicolson scheme has been chosen. Due to expected complexity on the boundary of the tumor, since fine grid multigrid schemes allow for high resolution, a V-cylce multigrid scheme with a uniform rectangular grid will solve the discretized system.

4 Phase I: Discretization

4.1 Spatial Discretization

For a $N_x \times N_y$ size grid, we have ϕ_T, ϕ_D, μ, and n defined to be cell centered (blue). Ghost cells are created to handle boundary conditions of the cell centered terms (grey). $\boldsymbol{u_S}$ is defined on the north-south and east-west centered cell edges (black). $\boldsymbol{u_S}$ is a vector defined in the x and y directions with the east-west being the x and north-south being the y. The boundary conditions found in [5] are as follows: $\mu = p = 0$, $n = 1$, and $\zeta \cdot \nabla \phi_T = \zeta \cdot \nabla \phi_D = 0$.

Figure 3: Grid Representation



4.1.1 Cell-centered Discretization: Finite Difference

Assuming a rectangular tissue domain with uniform grid points, the Laplacian operator and the Laplacian with non-constant diffusivity/mobility are approximated to second order (21) and (22), respectively.

$$\triangle_d \phi_{i,j} = \frac{\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1} - 4\phi_{i,j}}{h^2} \tag{22}$$

$$\begin{aligned} \nabla_d \cdot (m \nabla_d \phi)_{i,j} = & \frac{A_x m_{i+\frac{1}{2},j}(\phi_{i+1,j} - \phi_{i,j}) + A_x m_{i-\frac{1}{2},j}(\phi_{i,j} - \phi_{i-1,j})}{h^2} \\ & + \frac{A_y m_{i,j+\frac{1}{2}}(f_{i,j+1} - f_{i,j}) + A_y m_{i,j-\frac{1}{2}}(f_{i,j} - f_{i,j-1})}{h^2}, \end{aligned} \tag{23}$$

where A_x and A_y are averaging operators.

4.1.2 Cell-edge discretization: WENO Scheme

In the discretization of the mathematical model in section 3.1, the convection term is discretized separate from the other terms. Spatially since ϕ is cell-centered and $\boldsymbol{u_S}$ is edge-centered velocity, a third-order upwind WENO approximation is used.

$$\nabla_d \cdot (\boldsymbol{u_S} \phi)_{i,j} = \nabla_d \cdot \boldsymbol{f}_{i,j} = \frac{f^{ew}_{i+\frac{1}{2},j} - f^{ew}_{i-\frac{1}{2},j}}{h} + \frac{f^{ns}_{i,j+\frac{1}{2}} - f^{ns}_{i,j-\frac{1}{2}}}{h}, \tag{24}$$

where $f = (f^{ew}, f^{ns})$ is the numerical upwind flux determined by

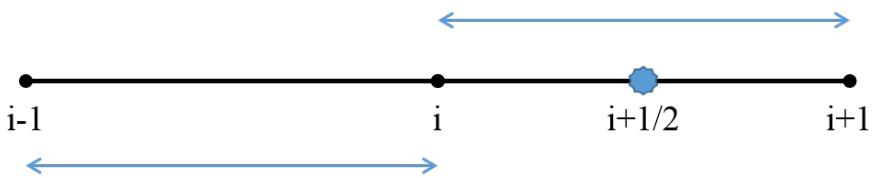
$$f^{ew}_{i+\frac{1}{2},j} = u^{ew}_{S\ i+\frac{1}{2},j} W^{ew}_{i+\frac{1}{2},j}(\phi) \tag{25}$$

$$f^{ns}_{i,j+\frac{1}{2}} = u^{ns}_{S\ i,j+\frac{1}{2}} W^{ns}_{i,j+\frac{1}{2}}(\phi) \tag{26}$$

and $W^{ew}_{i+\frac{1}{2},j}(\phi)$ is the upwind WENO reconstruction of ϕ on the east-west edges and $W^{ns}_{i+\frac{1}{2},j}(\phi)$ the reconstruction of ϕ for the north-south edges.

$W^{ew}_{i+\frac{1}{2},j}(\phi)$ is constructed by taking a linear combination of the interpolation of the data from the $i^{th}, i+1^{st}$, and $i-1^{st}$ points when the upwinding direction is i and the $i+1^{st}, i^{th}$, and $i+2^{nd}$ points when the upwinding direction is i+1. With full description in [1].

Figure 4: Representation of WENO stencils with i-direction upwinding



4.2 Temporal Discretization: Crank-Nicolson

If we begin with a partial derivative equation in two dimensions

$$\frac{\partial u}{\partial t} = F(t, x, y, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial y^2}), \tag{27}$$

the Crank-Nicolson discretization is a 2^{nd} order convergence in time based on the trapezoidal rule as follows:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{1}{2}[F_i^{n+1}(t, x, y, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial^2 u}{\partial x^2}\frac{\partial^2 u}{\partial y^2}) + F_i^n(t, x, y, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial y^2})]. \tag{28}$$

Temporally the diffusion term, or the excessive adhesion term, in equation (14) will be treated explicitly. Numerical tests from Wise et al [5] result in poor multigrid convergence when this term is treated implicitly.

4.3 Complete Discretization

Once discretized we have a system of equations as such:

$$\begin{aligned} \phi_{Ti,j}^k - \phi_{Ti,j}^{k-1} &= \frac{sM}{2}[\nabla_d(\phi_T^k \nabla_d \mu^k)_{i,j} + \nabla_d(\phi_T^{k-1}\nabla_d \mu^{k-1})_{i,j}] \\ -\frac{s}{2}[\nabla_d \cdot (\boldsymbol{u_S^k}\phi_T^k)_{i,j} + \nabla_d \cdot (\boldsymbol{u_S^{k-1}}\phi_T^{k-1})_{i,j}] &+ \frac{s}{2}[S_{Ti,j}^k + S_{Ti,j}^{k-1}] \end{aligned} \tag{29}$$

$$\mu_{i,j}^k = f'(\phi_{Ti,j}^k) - \varepsilon^2 \triangle_d \phi_{Ti,j}^k \tag{30}$$

$$\begin{aligned} \phi_{Di,j}^k - \phi_{Di,j}^{k-1} &= \frac{sM}{2}[\nabla_d(\phi_D^k \nabla_d \mu^k)_{i,j} + \nabla_d(\phi_D^{k-1}\nabla_d \mu^{k-1})_{i,j}] \\ -\frac{s}{2}[\nabla_d \cdot (\boldsymbol{u_S^k}\phi_D^k)_{i,j} + \nabla_d \cdot (\boldsymbol{u_S^{k-1}}\phi_D^{k-1})_{i,j}] &+ \frac{s}{2}[S_{Di,j}^k + S_{Di,j}^{k-1}] \end{aligned} \tag{31}$$

$$\begin{aligned} 0 &= \nabla_d \cdot (\kappa(\phi_T^k, \phi_D^k)_{i,j}\nabla p) + S_{Ti,j} \\ -\frac{\gamma}{\varepsilon}\nabla_d \cdot (\kappa(\phi_T^{k-1}, \phi_D^{k-1})\mu^{k-1}\nabla_d \phi_T^{k-1})_{i,j} & \end{aligned} \tag{32}$$

$$0 = \nabla_d \cdot (D(\phi_T^k)\nabla_d n^k)_{i,j} + n_{i,j}^k[(\phi_{Ti,j} - \phi_{Di,j}^k) + S_{ci,j}^k] - n_c S_{ci,j}^k \tag{33}$$

where,

$$S_{ci,j}^k := v_p^H(1 - Q(\phi_{Ti,j})) + v_p^T Q(\phi_{Ti,j}^k) \tag{34}$$

4.4 Implementation

Diffusion Terms

There are two MATLAB functions used to calculate the diffusion terms in the system of equations. When ϕ_* and μ are known, from a previous time step or fixed point iterate, the first function diffusion(h,varargin) takes input either (h,phi) or (h,m,phi), representing constant and non-constant diffusivity/mobility, respectively. This function calculates

the entire diffusion term matrix. When ϕ_* and μ are unknown, the second function, diffusion2(h,Nx,Ny,indicator), creates the matrices that are to be multiplied by the unknown ϕ_* and μ matrices. If there is constant diffusivity/mobility the matrix created is a matrix of size $(N_x N_y)^2$ with -4 on the diagonal and 1 on the superdiagonal, subdiagonal, $N_x + 1^{st}$ superdiagonal, and $N_x + 1^{st}$ subdiagonal. When there is non-constant diffusivity/mobility, the function creates two matrices for the averaging operators, one in the x-direction and the other in the y and four matrices for the phi term with either 1 or -1 on the sub or super diagonal and the opposite on the diagonal.

Advection Terms

Using the method described in section (4.3), the MATLAB function for calculating the advection terms, advection(h,u_ew,u_ns,phi), inputs the velocity function, $\boldsymbol{u_S}$ which is cell-edge centered and ϕ_* which is cell-centered and returns the advection matrix. Within the advection function, ϕ is input into the WENO(phi,indicator) function which reconstructs ϕ to the cell-edges with the third-order WENO scheme for both upwind in the i^{th} and $i + 1^{st}$ directions. Once the reconstruction returns to the advection function, the two matrices returned from the WENO function are combined to with the correct upwinding direction determined by the sign of the individual elements of $\boldsymbol{u_S}$. The function this returns the advection matrix using equation (24).

Validation of Code for Discretization

To test that the code for approximating the diffusion terms is correct, a test was done with $\phi(x, y) = \frac{(x-20)^2}{1.1} + (y-20)^2 - 4$ which is the initial value of ϕ_T provided in [5]. For simplicity, we define $\mu = \frac{(x-20)^3}{1.1} + (y-20)^3 - 4$ and $\boldsymbol{u_S} = [x + y, 2 * (x + y)]$.

The values for the exact matrix are determined by plugging in the corresponding grid values used to approximate the values. The truncation relative errors are calculated by taking the 2-norm of the difference of the approximate and exact matrices divided by the 2-norm of the exact matrix. The results are below:

Function	Relative Error (h=10/32)
diffusion(constant)	2.5121013001e-13
diffusion(non-constant)	9.8564532777e-05
advection	3.3281851089

When comparing the relative error of the two diffusion equations, the difference in the amount is conjectured to be due to the large degree which the numbers are raised to in the non constant diffusion. Further validation is required with the incorporation of the Crank-Nicolson temporal scheme with a simpler analytic function to fully understand the truncation error and to make sure that the expected order of accuracy is being achieved. Also, for the advection validation, the relative error is rather large, this is believed to be due to an error in removing the ghost cells in the WENO scheme. Correction of this error is in review and must be fixed before further validation is to take place.

5 Phase II: Solving the System

5.1 Separation into Source and Operator Terms

In order to solve the system of discretized equations listed in section 4.3, a fixed point algorithm will be used. First, we distinguish between the terms of the system in the current time step that are at the current fixed point iterate and those at the previous fixed point iterate. Using equations (29)-(34) we rewrite the equations in terms of $\widetilde{\boldsymbol{N}}$:= a vector of operator terms and $\widetilde{\boldsymbol{F}}$:= a vector of source terms such that $\widetilde{\boldsymbol{N}} = \widetilde{\boldsymbol{F}}$ = 'our system'. The selection of $\widetilde{\boldsymbol{N}}$ and $\widetilde{\boldsymbol{F}}$ can be found in [5]. $\boldsymbol{F}$ is a function of the dependent variables of the system with all of the values of these variables coming from a previous time step or previous fixed point iterate. $\boldsymbol{N}$, however, is also a function of the dependent variables of the system, but some of these values are from a previous iterate while the rest represent the current time step and iterate, which must be solved for.

5.2 Fixed Point Algorithm

Define $\boldsymbol{\Psi^*} = \{\phi_T^*, \phi_D^*, \mu^*, p^*, n^*\}$

$\boldsymbol{\Psi^{k,0}} \leftarrow \boldsymbol{\Psi^{k-1}}$

Solve loop:

for $m = 1$ **until** m_{max}

$\boldsymbol{\Psi^{k,m}}$=**Solve**$(\boldsymbol{N}(\boldsymbol{\cdot}, \boldsymbol{\Psi^{k,m-1}}) = \boldsymbol{F}(\boldsymbol{\Psi^{k,m-1}}, \boldsymbol{\psi^{k-1}}))$

if $\|\boldsymbol{F}(\boldsymbol{\Psi^{k,m}}, \boldsymbol{\Psi^{k-1}}) - \boldsymbol{N}(\boldsymbol{\Psi^{k,m}}, \boldsymbol{\Psi^{k,m}})\| < tol$

exit Solve loop

end for solve loop

$\boldsymbol{\Psi^{k}} \leftarrow \boldsymbol{\Psi^{k,m}}$

where k is the time step and m is the iterate of the fixed point algorithm

5.3 Multigrid: V-cycle (Future Work) [2]

5.3.1 Jacobi Smoother

The first attempt for the **Solve** routine in section (5.2) is a Newton's method with a locally linearized Jacobian calculated. Within the operator terms $\boldsymbol{N}$, in order to linearize the system, every variable is defined at the previous iterate except for the (i,j) term. It is anticipated by [1] that this smoother has slow convergence, thus when implementing the multigrid algorithm, the Jacobi smoother will be modified to the Gauss-Seidel smoother.

5.3.2 Multigrid

The second attempt for the **Solve** routine is through multigrid. Following the methods used in Wise et al. [5], when solving the discretized system of equations a V-cycle multigrid method with a Gauss-Seidel smoother will be used.

At each grid level

Pre-Smoothing: Gauss-Seidel Scheme

When solving $A\mathbf{u} = \mathbf{f}$

- Decompose $A = L + U$ where L is lower triangular and U is strictly upper triangular
- Solve the system $L(\boldsymbol{u}^{k+1}) = \boldsymbol{f} - U\boldsymbol{x}^k$

if coarsest level

solve $A\mathbf{u} = \mathbf{f}$

else

Restrict Residual

Correction for coarser grid

Update

end

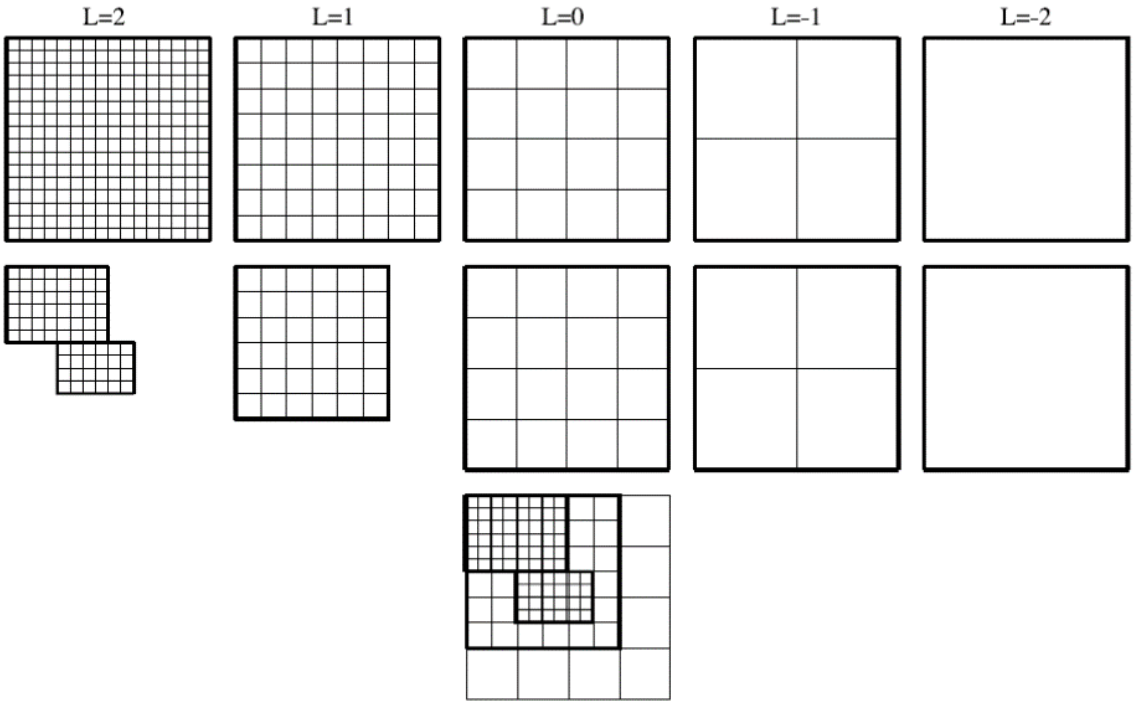
Post-Smoothing: Gauss-Seidel Scheme

Despite the usage of V-cycle multigrid in Wise et al. [5], according to Trottenberg et al. [4] the Gauss-Seidel smoothing on the V-cycle multigrid has level dependent convergence with worsening convergence with increased levels. This convergence is shown to improve significantly when there is additional line relaxation along the points adjacent to the boundary points or when a W-cycle multigrid algorithm is used. Thus, time permitting this line relaxation will be implemented or a W-cycle multigrid algorithm will be used to improve convergence.

5.3.3 Locally Refined Grid

Time permitting, the uniform square grids used in the multigrid algorithm will be replaced by locally refined grids where at each grid refinement, the grid is only refined in areas which require higher resolution as seen in Figure 2 from Wise et al 2011.

Figure 5: Uniform vs Locally Refined Grids [5]



5.3.4 Validation Methods

When testing the multigrid algorithm, two tests problems will be used which are similar to the mathematical model being used. The first (28) provides a simplistic version of a basic diffusion problem, where equation (29), a biharmonic diffusion PDE, contains the stiff fourth-order term similar to the diffusion term in the mathematical model for solid tumor growth. Both equations will be discretized just as those in the mathematical model for solid tumor growth.

1. The heat equation:
$$\frac{\partial u}{\partial t} = \nabla^2 u \tag{35}$$
2. 4th Order PDE
$$\frac{\partial u}{\partial t} = -\nabla^4 u \tag{36}$$

5.3.5 Verification

Using the initial conditions and parameters provided in Wise et al.[5], the results of project are anticipated to be that which is seen in Figure 3 and 4, where Figure 3 shows the simulated growth of the tumor at each time step and Figure 4 shows the initial tumor and the final results when t=200. These results are found starting with a coarse grid of 64^2, the finest grid of 512^2, and a time step of $1\text{x}10^{-2}$ and spatial step of $40/512$. Each contain the same

biophysical parameters with varying γ values. All images for comparison are provided from Wise et al 2011.

Figure 6: Projected Results with $\gamma = -0.1$ [5]

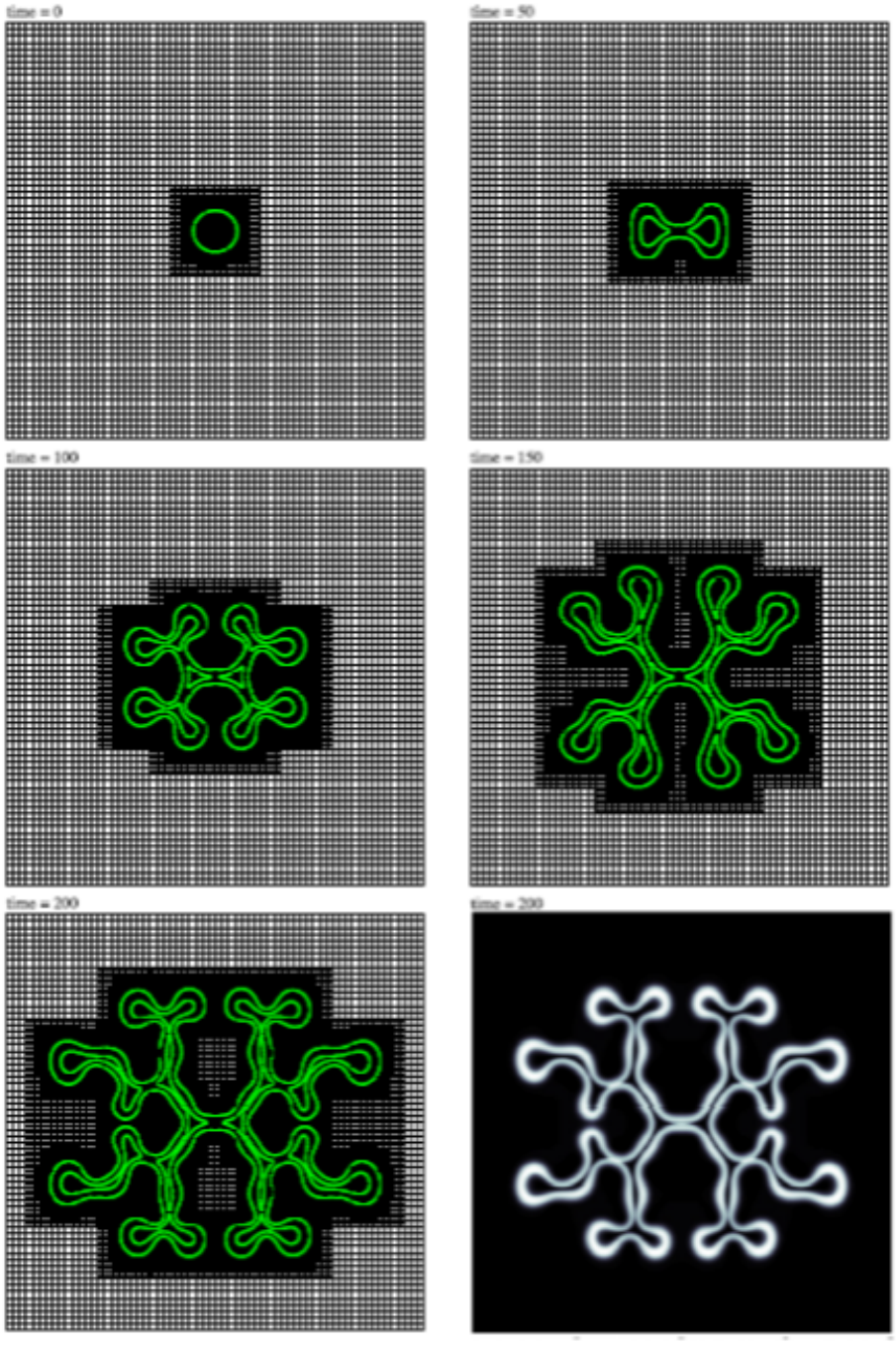
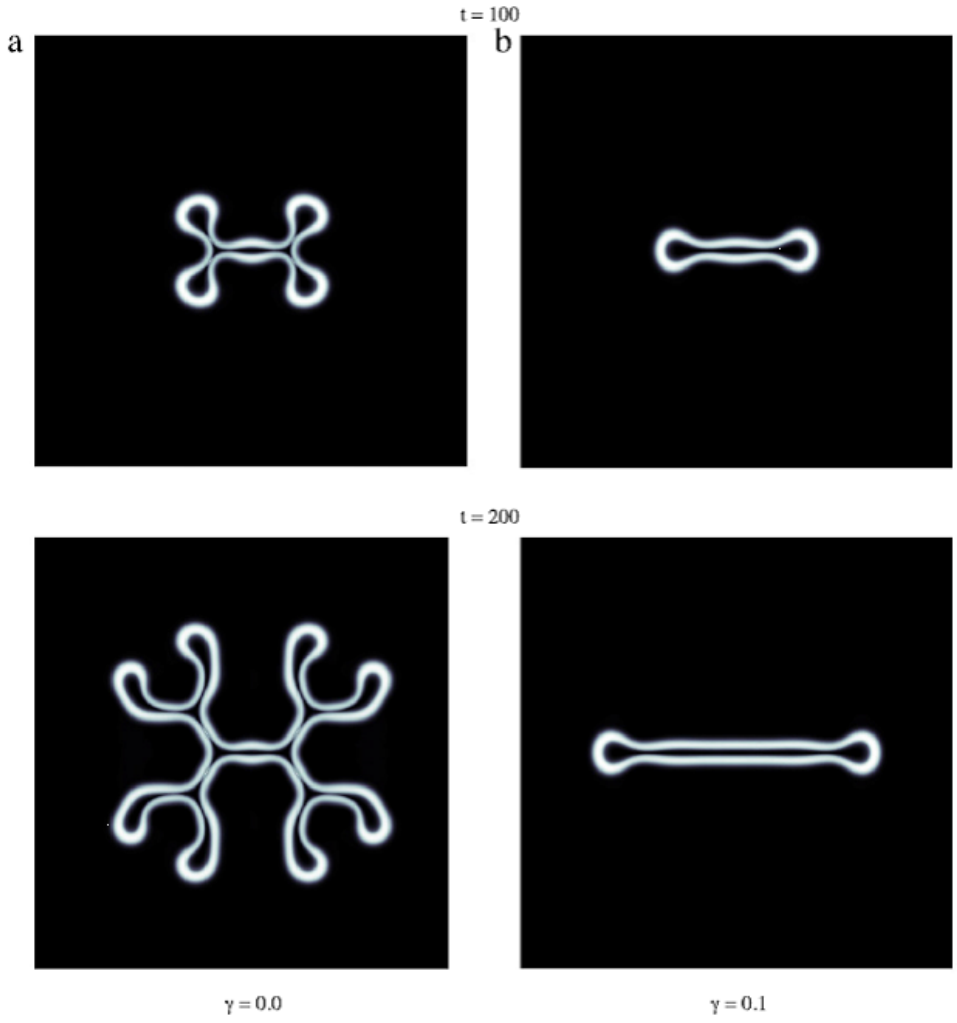


Figure 7: Projected Results with varying initial γ values [5]



6 Timeline and Milestones

Original Timeline:

October-November:

Discretize the mathematical model using the techniques described in sections 3.2.1-3.2.3 and use an implicit MATLAB solver to solve the discretized system to gain an initial result. Due to the anticipated length in computation time a smaller time progression will be used in comparison for the time progression with the use of the multigrid algorithm.

December-January:

Have the multigrid algorithm with uniform grid found in section 3.2.4 programmed and tested on validation problems found in section 4.

Modified Timeline:

October-November:

Discretize the mathematical model using the techniques described in sections 3.2.1-3.2.3.

December-January:

- Further validate the discretization functions
- Use the locally linearized smoother in section (5.3) to get an initial solution. **Note**: This change in the timeline is due to the fact that when exploring other implicit solvers, there is a requirement to calculate the global Jacobian matrix of the system. We determined this computation adds unnecessary complexity to the project when this Jacobian would not be used in our multigrid algorithm, which is the ultimate goal of the project.
- Have the multigrid algorithm with uniform grid found in section 3.2.4 programmed and tested on validation problems found in section 4.

February-March:

Combine the multigrid algorithm programmed at the end of December with the discretized 2D solid tumor growth mathematical model and apply the verification problems found in section 5.

April-May:

Finalize the results and prepare the final presentation and report.

Time Permitting:

- Modify the V-cycle multigrid algorithm to a W-cycle multigrid algorithm or add additional line relaxation along the points adjacent to the boundary points to improve multigrid convergence
- Extend the multigrid algorithm from uniform grids to the locally refined grids and have it tested on 2D model.
- Extend the 2D mathematical model to the 3D solid tumor growth model and use the multigrid algorithm to solve the system.

7 Deliverables

- Weekly Reports
- Proposal Presentation and Report
- Mid-Year Presentation and Report

- Final Presentation and Report
- Matlab Code for discretized solid tumor growth model
- Matlab Code for uniform multigrid algorithm
- Documentation for tumor growth model simulation

References

[1] V. Cristini and J.S. Lowengrubb. *Multiscale Modeling of Cancer: An Integrated Experimental and Mathematical Approach*. Cambridge University Press, 2010.

[2] Howard C. Elman, David J. Silvester, and Andrew J. Wathen. *Finite Elements and Fast Iterative Solvers*. Oxford University Press, second edition, 2014.

[3] University of Oxford: Department of Computer Science. Cell-based chaste: a multiscale computational framework for modelling cell populations, 2014.

[4] U. Trottenberg, Cornelius Oosterlee, and Anton Schuller. *Multigrid*. Academic Press, 2001.

[5] S.M. Wise, J.S. Lowengrubb, and V. Cristini. An adaptive multigrid algorithm for simulating solid tumor growth using mixture models. *Mathematical and Computer Modelling*, 53:1–20, 2011.