# Simulating Solid Tumor Growth using Multigrid Algorithms: Mid-Year Presentation

Asia Wyatt

Applied Mathematics, Statistics, and Scientific Computation Program

Advisor: Doron Levy, PhD

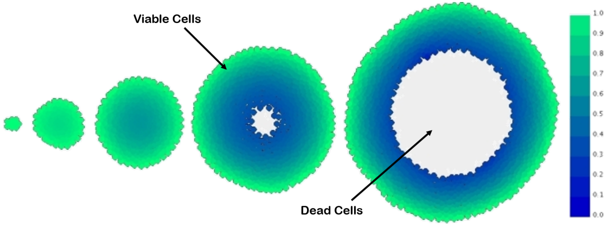
Department of Mathematics/CSCAMM

University of Maryland, College Park

December 4, 2014

# Overview

# Project Goal




**Tumor Growth Over Time**[1]

Simulate solid tumor growth in two and, time permitting, three dimensions with a Cahn-Hilliard-type convection-reaction-diffusion mathematical model using multigrid algorithms as done in Wise, Lowengrubb and Cristini 2011.

[1] University of Oxford: Department of Computer Science (2014). *Cell-based Chaste: a multiscale computational framework for modelling cell populations.*

# Mathematical Model

## Model Parameters

# Mathematical Model

## Model Parameters[2]

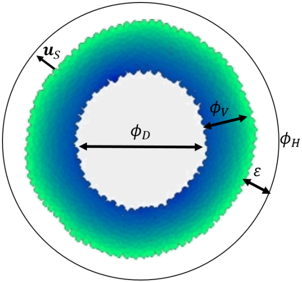
- $\phi_V$: volume fraction of viable tissue
- $\phi_D$: volume fraction of dead tissue
- $\phi_H$: volume fraction of healthy tissue
- $\boldsymbol{u}_S$: tissue velocity
- $\varepsilon$: thickness of interface between healthy and tumoral tissue
- $p$: cell-to-cell (solid) pressure
- $n$: nutrient concentration

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

# Mathematical Model

## Model Parameters[2]

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




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

# Mathematical Model

If $\phi$ represents the tissue volume fraction, then

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

$$\frac{\partial \phi_T}{\partial t} = M\nabla(\phi_T \nabla \mu) + S_T - \nabla \cdot (\boldsymbol{u_S} \phi_T) \tag{2}$$

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

$$0 = \nabla \cdot (D(\phi_T)\nabla n) + T_c(\phi_T, n) - n(\phi_T - \phi_D), \tag{4}$$

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.

# Mathematical Model

When looking strictly at equation (2), we have

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

# Mathematical Model

When looking strictly at equation (2), we have

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

a diffusion term

# Mathematical Model

When looking strictly at equation (2), we have

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

a diffusion term, a source term

# Mathematical Model

When looking strictly at equation (2), we have

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

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

# Methods

## Tissue Velocity Equations

$$\boldsymbol{u_S} = -\kappa(\phi_T, \phi_D)(\nabla p - \frac{\gamma}{\varepsilon}\nabla\phi_T) \quad (5)$$

with $\kappa$ 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,

$$\nabla \cdot \boldsymbol{u_S} = S_T \quad (6)$$

Combined we have a Poisson equation in $p$,

$$-\nabla \cdot (\kappa(\phi_T, \phi_D)\nabla p) = S_T - \nabla \cdot (\kappa(\phi_T, \phi_D)\frac{\gamma}{\varepsilon}\nabla\phi_T) \quad (7)$$

Thus, $p$ can be solved for independently then used to find $\boldsymbol{u_S}$ in equation (5).

# Methods

## Source Terms

# Methods

## Source Terms

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

$$S_T = nG(\phi_T)\phi_V - \lambda_L\phi_D, \tag{8}$$

where $\lambda_L$ is the rate at which the lysing of dead cells form water and

$$G(x) = \begin{cases} 1 & \text{if } \frac{3\varepsilon}{2} \leq \phi \\ \phi/\varepsilon - \frac{1}{2} & \text{if } \frac{\varepsilon}{2} < \phi < \frac{3\varepsilon}{2} \\ 0 & \text{if } \phi \leq \frac{\varepsilon}{2}. \end{cases} \tag{9}$$

# Methods

## Source Terms

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

$$S_T = nG(\phi_T)\phi_V - \lambda_L\phi_D, \tag{8}$$

where $\lambda_L$ is the rate at which the lysing of dead cells form water and

$$G(x) = \begin{cases} 1 & \text{if } \frac{3\varepsilon}{2} \leq \phi \\ \phi/\varepsilon - \frac{1}{2} & \text{if } \frac{\varepsilon}{2} < \phi < \frac{3\varepsilon}{2} \\ 0 & \text{if } \phi \leq \frac{\varepsilon}{2}. \end{cases} \tag{9}$$

Assuming that the net source of the dead cells $S_D$ is due to apoptosis and necrosis with rates $\lambda_A$ and $\lambda_N$, respectively, we have

$$S_D = (\lambda_A + \lambda_N \mathscr{H}(n_N - n))(\phi_T - \phi_D) - \lambda_L\phi_D, \tag{10}$$

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.

# Methods

## Nutrient Equation

$$0 = \nabla \cdot (D(\phi_T)\nabla n) + T_c(\phi_T, n) - n(\phi_T - \phi_D) \quad (11)$$

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,

$$D(\phi) = D_H(1 - Q(\phi_T)) + Q(\phi_T), \quad (12)$$

$$T_c(\phi_T, n) = (v_p^H(1 - Q(\phi_T)) + v_p^T Q(\phi_T))(n_c - n), \quad (13)$$

and

$$Q(\phi) = \begin{cases} 1 & \text{if } 1 \leq \phi \\ 3\phi^2 - 2\phi^3 \textbf{or} \phi & \text{if } 0 < \phi < 1 \\ 0 & \text{if } \phi \leq 0. \end{cases} \quad (14)$$

# Methods

## Coupled System

$$\frac{\partial \phi_T}{\partial t} = M\nabla(\phi_T \nabla \mu) + S_T - \nabla \cdot (\boldsymbol{u_S} \phi_T) \tag{15}$$

$$\frac{\partial \phi_D}{\partial t} = M\nabla(\phi_T \nabla \mu) + S_D - \nabla \cdot (\boldsymbol{u_S} \phi_D) \tag{16}$$

$$0 = \nabla \cdot (D(\phi_T)\nabla n) + T_c(\phi_T, n) - n(\phi_T - \phi_D) \tag{17}$$

where,

$$\mu = f'(\phi_T) - \varepsilon^2 \nabla^2 \phi_T \tag{18}$$

$$f(\phi) = \phi^2 (1-\phi)^2/2 \tag{19}$$

$$\boldsymbol{u_S} = -\kappa(\phi_T, \phi_D)(\nabla p - \frac{\gamma}{\varepsilon}\nabla \phi_T) \tag{20}$$

$$S_T = nG(\phi_T)\phi_V - \lambda_L \phi_D, \tag{21}$$

$$S_D = (\lambda_A + \lambda_N \mathscr{H}(n_N - n))(\phi_T - \phi_D) - \lambda_L \phi_D. \tag{22}$$

Thus, the system is complete.

# Methods

## Discretization

Discretize the mathematical model for solid tumor growth in 2D both spatially by finite difference and in time using an implicit Crank-Nicolson scheme due to 4th order diffusion term. Implement well known matrix solver to solve the final system of equations.

## Uniform Multigrid

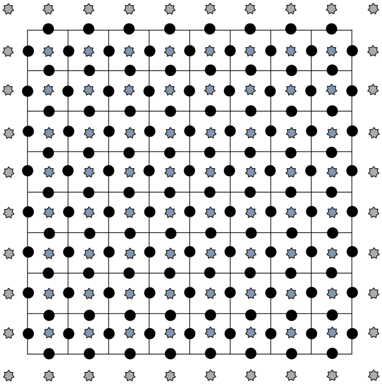
Due to expected complexity on boundary of tumor, fine grid multigrid schemes allow for high resolution. Thus, a multigrid algorithm with a uniform mesh will solve the discretized system.

# Phase I: Discretization

# Spatial Discretization: Finite Difference

# Spatial Discretization: Finite Difference

For a $N_x \times N_y$ size grid, we have $\phi_T, \phi_D, \mu$, and $n$ defined to be cell centered (blue). Ghost cells are created to handle boundary conditions of the cell centered terms (grey). $u_S$ is defined on the north-south and east-west centered cell edges (black). The boundary conditions found in Wise, Lowengrubb and Cristini 2011 are as follows: $\mu = p = 0$, $n = 1$, and $\zeta \cdot \nabla \phi_T = \zeta \cdot \nabla \phi_D = 0$



# Spatial 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 (18) and (19), respectively.

$$\triangle_d \phi_{i,j} = \frac{\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1} - 4\phi_{i,j}}{h^2} \tag{23}$$

$$\begin{aligned} \nabla_d \cdot (m \nabla_d \phi)_{i,j} = {} & \frac{A_x m_{i+\frac{1}{2},j}(\phi_{i+1,j} - \phi_{i,j}) + A_x m_{i-\frac{1}{2},j}(\phi_{i,j} - \phi_{i-1,j})}{h^2} \\ & + \frac{A_y m_{i,j+\frac{1}{2}}(\phi_{i,j+1} - \phi_{i,j}) + A_y m_{i,j-\frac{1}{2}}(\phi_{i,j} - \phi_{i,j-1})}{h^2}, \end{aligned} \tag{24}$$

where $A_x$ and $A_y$ are averaging operators

# Spatial Discretization: WENO Scheme

When discretizing the advection term a third-order upwind WENO approximation is used.

$$\nabla_d \cdot (\boldsymbol{u_S}\phi)_{i,j} = \nabla_d \cdot \boldsymbol{f}_{i,j} = \frac{f^{ew}_{i+\frac{1}{2},j} - f^{ew}_{i-\frac{1}{2},j}}{h} + \frac{f^{ns}_{i,j+\frac{1}{2}} - f^{ns}_{i,j-\frac{1}{2}}}{h}, \quad (25)$$

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

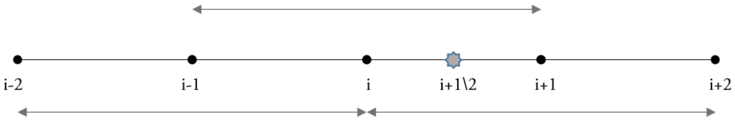
$$f^{ew}_{i+\frac{1}{2},j} = u^{ew}_{S\ \ i+\frac{1}{2},j} W^{ew}_{i+\frac{1}{2},j}(\phi) \quad (26)$$

$$f^{ns}_{i,j+\frac{1}{2}} = u^{ns}_{S\ \ i,j+\frac{1}{2}} W^{ns}_{i,j+\frac{1}{2}}(\phi) \quad (27)$$

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

# Spatial Discretization: WENO Scheme

$W^{ew}_{i+\frac{1}{2},j}(\phi)$ is constructed by interpolating the data from the $i^{th}$, $i+1^{st}$, $i+2^{nd}$, $i-1^{st}$, and $i-1^{st}$



# Temporal Discretization: Crank-Nicolson

If we begin with a partial derivative equation in two dimensions

$$\frac{\partial u}{\partial t} = F(t, x, y, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial y^2}), \tag{28}$$

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

$$\begin{aligned} \frac{u_i^{n+1} - u_i^n}{\Delta t} &= \frac{1}{2}[F_i^{n+1}(t, x, y, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial^2 u}{\partial x^2} \frac{\partial^2 u}{\partial y^2}) \\ &+ F_i^n(t, x, y, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial y^2})] \end{aligned} \tag{29}$$

# MATLAB Implementation

Once discretized we have a system of equations as such:

# MATLAB Implementation

Once discretized we have a system of equations as such:

$$\begin{aligned}\phi_{Ti,j}^{k} - \phi_{Ti,j}^{k-1} = \frac{sM}{2}[\nabla_d(\phi_T^k \nabla_d \mu^k)_{i,j} + \nabla_d(\phi_T^{k-1} \nabla_d \mu^{k-1})_{i,j}] \\ -\frac{s}{2}[\nabla_d \cdot (\boldsymbol{u_S^k} \phi_T^k)_{i,j} + \nabla_d \cdot (\boldsymbol{u_S^{k-1}} \phi_T^{k-1})_{i,j}] + \frac{s}{2}[S_{Ti,j}^k + S_{Ti,j}^{k-1}]\end{aligned} \tag{30}$$

# MATLAB Implementation

Once discretized we have a system of equations as such:

$$\phi_{Ti,j}^{k} - \phi_{Ti,j}^{k-1} = \frac{sM}{2}[\nabla_d(\phi_T^k \nabla_d \mu^k)_{i,j} + \nabla_d(\phi_T^{k-1}\nabla_d \mu^{k-1})_{i,j}]$$
$$-\frac{s}{2}[\nabla_d \cdot (\boldsymbol{u_S^k}\phi_T^k)_{i,j} + \nabla_d \cdot (\boldsymbol{u_S^{k-1}}\phi_T^{k-1})_{i,j}] + \frac{s}{2}[S_{Ti,j}^k + S_{Ti,j}^{k-1}] \quad (30)$$

$$\mu_{i,j}^k = f'(\phi_{Ti,j}^k) - \varepsilon^2 \triangle_d \phi_{Ti,j}^k \quad (31)$$

$$\phi_{Di,j}^{k} - \phi_{Di,j}^{k-1} = \frac{sM}{2}[\nabla_d(\phi_D^k \nabla_d \mu^k)_{i,j} + \nabla_d(\phi_D^{k-1}\nabla_d \mu^{k-1})_{i,j}]$$
$$-\frac{s}{2}[\nabla_d \cdot (\boldsymbol{u_S^k}\phi_D^k)_{i,j} + \nabla_d \cdot (\boldsymbol{u_S^{k-1}}\phi_D^{k-1})_{i,j}] + \frac{s}{2}[S_{Di,j}^k + S_{Di,j}^{k-1}] \quad (32)$$

$$0 = \nabla_d \cdot (\kappa(\phi_T^k, \phi_D^k)_{i,j} \nabla p) + S_{Ti,j}$$
$$-\frac{\gamma}{\varepsilon}\nabla_d \cdot (\kappa(\phi_T^{k-1}, \phi_D^{k-1})\mu^{k-1}\nabla_d \phi_T^{k-1})_{i,j} \quad (33)$$

$$0 = \nabla_d \cdot (D(\phi_T^k)\nabla_d n^k)_{i,j} + n_{i,j}^k[(\phi_{Ti,j} - \phi_{Di,j}^k) + S_{ci,j}^k] - n_c S_{ci,j}^k \quad (34)$$

where,

$$S_{ci,j}^k := v_p^H(1 - Q(\phi_{Ti,j})) + v_p^T Q(\phi_{Ti,j}^k) \quad (35)$$

# MATLAB Implementation

## Fixed Point Method

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

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

Solve loop:

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

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

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

**exit** Solve loop

**end for** solve loop

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

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

# MATLAB Implementation

## Fixed Point Method

Using equations (30)-(35) we have $\widetilde{\boldsymbol{N}}$ := a vector of operator terms and $\widetilde{\boldsymbol{F}}$ := a vector of source terms such that $\widetilde{\boldsymbol{N}} = \widetilde{\boldsymbol{F}}$.
For example: In equation (30)

$$\phi^k_{Ti,j} - \phi^{k-1}_{Ti,j} = \frac{sM}{2}[\nabla_d \cdot (\phi^k_T \nabla_d \mu^k)_{i,j} + \nabla_d(\phi^{k-1}_T \nabla_d \mu^{k-1})_{i,j}] \\ -\frac{s}{2}[\nabla_d \cdot (\boldsymbol{u}^{\boldsymbol{k}}_{\boldsymbol{S}} \phi^k_T)_{i,j} + \nabla_d \cdot (\boldsymbol{u}^{\boldsymbol{k-1}}_{\boldsymbol{S}} \phi^{k-1}_T)_{i,j}] + \frac{s}{2}[S^k_{Ti,j} + S^{k-1}_{Ti,j}]$$

So, as decided in Wise, Lowengrubb and Cristini 2011

$$\widetilde{N}^{(1)}_{i,j} = \phi^k_{Ti,j} - \tfrac{sM}{2}\nabla_d \cdot (\phi^k_T \nabla_d \mu^k)_{i,j}$$

and

$$\widetilde{F}^{(1)}_{i,j} = \phi^{k-1}_{Ti,j} + \tfrac{sM}{2}(\nabla_d(\phi^{k-1}_T \nabla_d \mu^{k-1})_{i,j}) + \tfrac{s}{2}[S^k_{Ti,j} + S^{k-1}_{Ti,j}] - \tfrac{s}{2}[\nabla_d \cdot (\boldsymbol{u}^{\boldsymbol{k}}_{\boldsymbol{S}} \phi^k_T)_{i,j} + \nabla_d \cdot (\boldsymbol{u}^{\boldsymbol{k-1}}_{\boldsymbol{S}} \phi^{k-1}_T)_{i,j}]$$

# MATLAB Implementation

## Fixed Point Method

Thus,

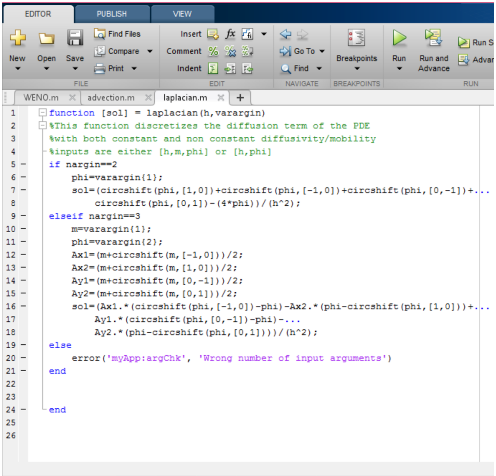
$$N^{(1)}_{i,j}(\boldsymbol{\Psi}^{\boldsymbol{k,m}}, \boldsymbol{\Psi}^{\boldsymbol{k,m-1}}) = \phi^{k,m}_{Ti,j} - \frac{sM}{2}\nabla_d \cdot (\phi^{k,m}_T \nabla_d \mu^{k,m})_{i,j}$$

and

$$F^{(1)}_{i,j}(\boldsymbol{\Psi}^{\boldsymbol{k,m-1}}, \boldsymbol{\Psi}^{\boldsymbol{k-1}}) = \phi^{k-1}_{Ti,j} + \frac{sM}{2}(\nabla_d(\phi^{k-1}_T \nabla_d \mu^{k-1})_{i,j}) + \frac{s}{2}[S^{k,m-1}_{Ti,j} + S^{k-1}_{Ti,j}] - \frac{s}{2}[\nabla_d \cdot (\boldsymbol{u}^{\boldsymbol{k}}_{\boldsymbol{S}} \phi^{k,m-1}_T)_{i,j} + \nabla_d \cdot (\boldsymbol{u}^{\boldsymbol{k-1}}_{\boldsymbol{S}} \phi^{k-1}_T)_{i,j}]$$

All choices of source terms and operator terms are from Wise, Lowengrubb and Cristini 2011. In MATLAB all of the source terms are calculated as full matrices and the operator terms are implemented as matrix vector products. Thus, the **Solve** command must solve a nonlinear system of size $5 \times N_x \times N_y$.

# MATLAB Implementation




Functions Coded:

- Diffusion terms
- WENO scheme
- Advection terms
- Overall system (partial)
- Fixed point Algorithm (partial)

# Phase II: Solving the System

# Solvers

Solvers explored:

- Newton's Method
- Broyden's Method
- Picard Method

**Issues:** There is a need to calculate the Jacobian of the nonlinear system

**Solution:** Use the method described as the smoother in Cristini and Lowengrubb 2010 which calculates the Jacobian of the locally linearized system. This smoother will be used in the multigrid algorithm as well.

# Timeline

# Timeline

November 30:

Discretize the mathematical model and use an implicit MATLAB solver to get an initial solution

January 31:

Have an initial uniform multigrid algorithm programmed and tested on verification problems.

# Timeline

November 30:

Discretize the mathematical model and use an implicit MATLAB solver to get an initial solution

# Timeline

November 30:

Discretize the mathematical model and use an implicit MATLAB solver to get an initial solution

January 31:

- Use locally linearized smoother in Cristini and Lowengrubb 2010 to get an initial solution
- Have an initial uniform multigrid algorithm programmed and tested on verification problems.

# Timeline

November 30:

Discretize the mathematical model and use an implicit MATLAB solver to get an initial solution

January 31:

- Use locally linearized smoother in Cristini and Lowengrubb 2010 to get an initial solution
- Have an initial uniform multigrid algorithm programmed and tested on verification problems.

March 31:

Combine uniform multigrid algorithm and discretized 2D solid tumor growth mathematical model

# Timeline

November 30:

Discretize the mathematical model and use an implicit MATLAB solver to get an initial solution

January 31:

- Use locally linearized smoother in Cristini and Lowengrubb 2010 to get an initial solution
- Have an initial uniform multigrid algorithm programmed and tested on verification problems.

March 31:

Combine uniform multigrid algorithm and discretized 2D solid tumor growth mathematical model

Time Permitting:

- Extend uniform multigrid algorithm to the locally refined multigrid and have it tested on 2D model.
- Extend the multigrid algorithms to the 3D solid tumor growth model.

# Deliverables

1. Proposal Presentation and Report
2. Mid-Year Presentation and Report
3. Final Presentation and Report
4. Matlab Code for implicit solver for discretized solid tumor growth model
5. Matlab Code for uniform multigrid algorithm
6. Documentation for tumor growth model simulation

# Deliverables

1. Proposal Presentation and Report
2. Mid-Year Presentation and Report
3. Final Presentation and Report
4. Matlab Code for implicit solver for discretized solid tumor growth model
5. Matlab Code for uniform multigrid algorithm
6. Documentation for tumor growth model simulation

# References

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

Oxford: Department of Computer Science, University of (2014). *Cell-based Chaste: a multiscale computational framework for modelling cell populations*.

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

# Questions?