Computational Mechanics and Multi-Physics

Doctor of Philosophy

Debabrata Auddya

auddya@wisc.edu

Beginning 6th June 2018

Contents

Wednesday, 6 June 2018

1 A mathematical derivation of the free energy considering adhesion

0.1 A phase field formulation for cell growth, division and contact considering cell adhesion

Uptil previous literature, and extensive study has been perfomed to model a phase field representation of scalar fields. Several terms are considered for this treatment such as phase segregation of cell interior from the extra cellular matrix, intercellular repulsion (by penalising overlapping scalar field) and also considering certain boundary conditions which confine the cell evolution within a certain boundary. In this formulation we are going to consider adhesion between cells which essentially play a key role in soft packing.

0.2 Mathematical analysis of the revised model

Let $\Omega \in \mathbb{R}^2$ with a smooth boundary $\partial\Omega$. Scalar fields $c_k,\ k = 1, \ldots, N$ with $c_k \in [0, 1]$ serve to delineate the interior and exterior of the cell numbered k. Here, the interior of cell k is $\omega_k \subset \Omega$, where $\omega_k = \{\boldsymbol{X} \in \Omega | c_k(\boldsymbol{X}) = 1\}$. The exterior is $\Omega \backslash \omega_k$. The free energy density function is built up beginning with the following form:

$$\psi_1(c_k) = \alpha c_k^2 (c_k - 1)^2 + \frac{\kappa}{2} |\nabla c_k|^2 \tag{0.1}$$

where the double-well term, $f(c_k) = \alpha c_k^2 (c_k - 1)^2$, enforces segregation into ω_k and $\Omega \backslash \omega_k$. In Equation (0.1), the second term enforces a diffuse cell-matrix interface (the cell membrane) of finite thickness, where κ controls the interface thickness, and thereby the interfacial energy. For N cells in Ω, the above free energy density needs to be extended to model contact by adding a cell-cell repulsion term. The total free energy of the multi-cell aggregate is a functional $\Pi[\boldsymbol{c}]$, defined as

$$\begin{aligned} \Pi[\boldsymbol{c}] &:= \int_\Omega \psi(\boldsymbol{c}, \nabla c)\ \mathrm{d}V \\ &= \int_\Omega \left(\sum_{k=1}^{N} f(c_k) + \sum_{k=1}^{N} \frac{\kappa}{2} |\nabla c_k|^2 + \sum_{l \neq k} \sum_{k=1}^{N} \lambda c_k^2 c_l^2 + \sum_{l \neq k} \sum_{k=1}^{N} \gamma \nabla c_k \nabla c_l \right) \mathrm{d}V. \end{aligned} \tag{0.2}$$

Here, $\boldsymbol{c} = \{c_1, \ldots, c_N\}$,$\lambda$ is a penalty coefficient that enforces repulsion and γ is another penalty coefficient that enforces adhesion between any two cells k, l thus modelling cell contact.
Taking the variational derivative with respect to c_k in Equation (0.2) yields

$$\begin{aligned}\delta\Pi_k[\boldsymbol{c}; w] &= \frac{\mathrm{d}}{\mathrm{d}\epsilon}\int_\Omega \sum_{k=1}^{N}\left(f(c_k+\epsilon w) + \frac{\kappa}{2}|\nabla(c_k+\epsilon w)|^2 + \sum_{l\neq k}\lambda(c_k+\epsilon w)^2 c_l^2 + \sum_{l\neq k}\gamma\nabla(c_k+\epsilon w)\nabla c_l\right) \mathrm{d}V \\ &= \int_\Omega \left(w f'(c_k) - w\kappa\Delta c_k + w\sum_{l\neq k} 2\lambda c_k c_l^2 - \sum_{l\neq k}\gamma\nabla w\nabla c_l\right) dV + \int_{\partial\Omega} w\kappa\nabla c_k\cdot\boldsymbol{n} dS \\ &= \int_\Omega w\left(f'(c_k) - \kappa\Delta c_k + \sum_{l\neq k} 2\lambda c_k c_l^2 - \sum_{l\neq k}\gamma\nabla^2 c_l\right) dV + \int_{\partial\Omega} w\kappa\nabla c_k\cdot\boldsymbol{n}\, dS\end{aligned} \tag{0.3}$$

where $\boldsymbol{n}$ is the unit outward normal vector to $\partial\Omega$. The chemical potential of the k^{th} cell is identified as,

$$\mu_k = f'(c_k) - \kappa\Delta c_k + \sum_{l\neq k} 2\lambda c_k c_l^2 - \sum_{l\neq k}\gamma\nabla^2 c_l \tag{0.4}$$

At equilibrium, $\delta_k\Pi[\boldsymbol{c}; w] = 0$ for the k^{th} cell, yielding $\mu_k = 0$ in Ω, and $\kappa\nabla c_k\cdot\boldsymbol{n} = 0$ on $\partial\Omega$.
The following parabolic partial differential equation, popularly known as the Cahn-Hilliard equation [**?**], imposes the conserved dynamics that governs the delineation and growth of the $N-$cell agglomerate, and of repulsion between cell pairs:

$$\frac{\partial c_k}{\partial t} = -\nabla\cdot(-M\nabla\mu_k) + s_k \tag{0.5}$$

where the source term s_k has been introduced, and M is the mobility, assumed to be constant. The dynamics of the multi-cell soft packing problem is governed by Equation (0.5) with the thermodynamics prescribed by Equation (0.4) and boundary conditions $\kappa\nabla c_k\cdot\boldsymbol{n} = 0$, $c_k = 0$ on $\partial\Omega$ for $k = 1, \ldots N$.

Tuesday, 12th June 2018

1 deal.ii Code snippet of the free energy formulation

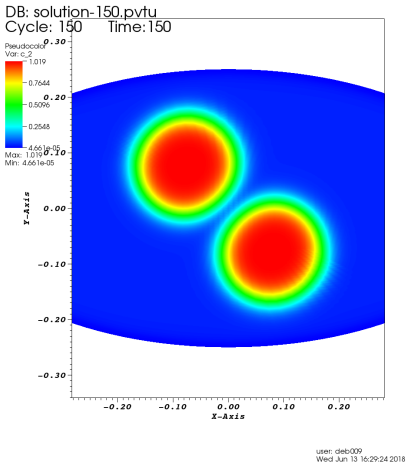
The code is modified to include the adhesion term in the free energy formulation.Observe here that the change is incorporated in the chemical potential residual term and the corresponding penalty term is included along with an optimised value of the gamma which controls the phenomenon between two cells. Later we will present different values of gamma for which the phase field simulation showed observable results.

```
else if(ck==1){
Sacado::Fad::DFad<double> dfdc = dFdC;
//add cross penalty terms to free energy
for (unsigned int cDof2=0; cDof2<CDOFs; cDof2++){
if (cDof2!=cDof) dfdc += 200*c[cDof][q]*c[cDof2][q]*c[cDof2][q];
}
//add surface buffer zone
if (cell->at_boundary()){
dfdc += 200*c[cDof][q]*1.0*1.0;
}
//
R[i] += fe_values.shape_value(i, q)*(mu[cDof][q] - dfdc)*fe_values.JxW(q);
//R[i] += fe_values.shape_value(i, q)*(mu[q] - dfdc - defMap.W[q] -
    pressure*defMap.divU[q])*fe_values.JxW(q);
for (unsigned int j = 0; j < dim; j++){
Sacado::Fad::DFad<double> Kjj= Kappa[j];
Sacado::Fad::DFad<double> kc_j= c_j[cDof][q][j]*Kjj; // Kjj*C_j
R[i] -= fe_values.shape_grad(i, q)[j]*(kc_j)*fe_values.JxW(q);
for (unsigned int cDof2=0; cDof2<CDOFs; cDof2++){
if (cDof2!=cDof){
R[i] -= 0.00085*fe_values.shape_grad(i,q)[j]*c_j[cDof2][q][j]*fe_values.JxW(q);
}
}
}
```

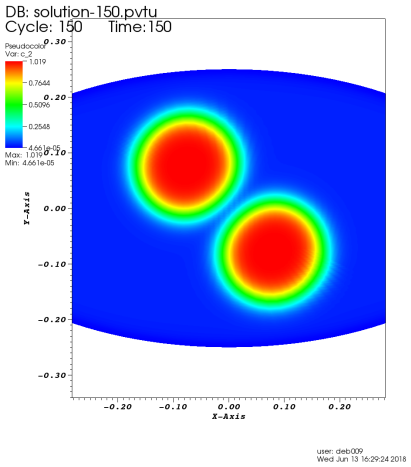
Wednesday, 13th June 2018

1 Phase field simulations results with adhesion term

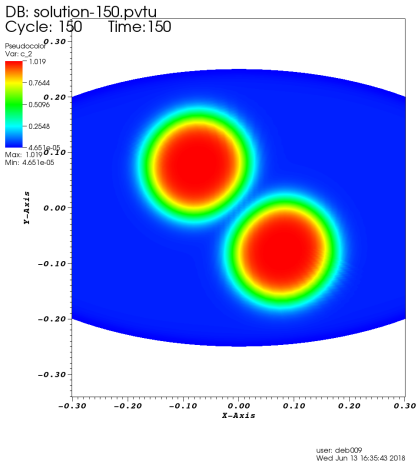
We present the phase field simulation results here considering the adhesion term and varying admissible values of gamma and the order parameters. All the images are captured at 150th iteration of the problem.



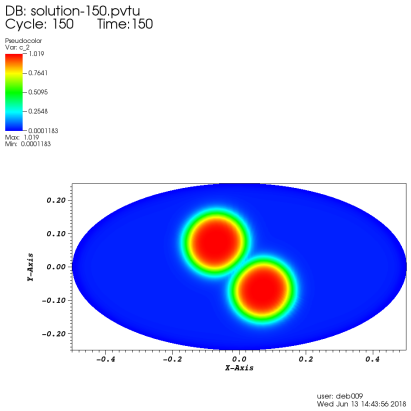
(a) For $\gamma = 0.0$



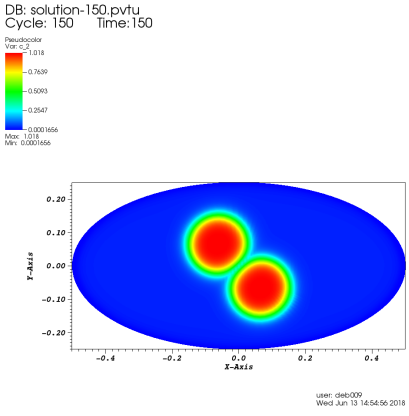
(b) For $\gamma = 1e\text{-}05$



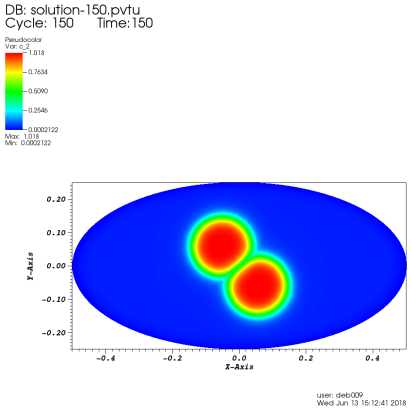
(c) For $\gamma = 1e\text{-}06$



(d) For $\gamma = 4e\text{-}04$



(e) For $\gamma = 6e\text{-}04$



(f) For $\gamma = 8e\text{-}04$

Figure 1:
A demonstration of adhesion for different values of γ.

Tuesday, 26th June 2018

1 Equivalence of Shape and Material Model

We present here a comparison between the shape model of the cell-packing formulation (essentially soft packing) and the Mooney-Rivlin material model. We present here a mathematical equivalence of the shape and material model for this soft packing problem to illustrate an empirical relationship between them.

0.3 The Shape Model

$$\begin{aligned}\Pi_s =& \sum_i \delta_i (I_i - I_i^{Ref}) \\ =& \delta_1 (I_1 - I_1^{Ref})^2 + \delta_2 (I_2 - I_2^{Ref})^2 \\ =& \delta_1 (\frac{\pi}{4} nR (\frac{R}{2})^3 - \frac{\pi R^4}{4}) + \delta_2 (\frac{\pi}{4} (nR)^3 (\frac{R}{2}) - \frac{\pi R^4}{4}) \\ =& (\frac{\pi R^4}{4})^2 (\delta_1 (\frac{1}{n^2} - 1)^2 + \delta_2 (n^2 - 1)^2) \end{aligned} \tag{0.6}$$

The shape model described above considers a two dimensional deformation of a circle into an ellipse under an incompressible situation. The major axis is stretched by a factor of $\mathbf{n}$(ellipticity) and the minor axis shrunk by the same factor. δ_1 and δ_2 are the shape constants for the problem.

0.4 The Mooney-Rivlin Material Model

$$\begin{aligned} W =& \lambda_a (\lambda_1^2 + \lambda_2^2 - 3) + \lambda_b (\lambda_1^2 \lambda_2^2 + \lambda_2^2 \lambda_3^2 + \lambda_3^2 \lambda_1^2 - 3) \\ =& \lambda_a (n^2 + (\frac{1}{n})^2 - 3) + \lambda_b (n^2 (\frac{1}{n})^2 - 3) \\ =& \lambda_a (n^2 + (\frac{1}{n})^2 - 3) - 2\lambda_b \end{aligned} \tag{0.7}$$

This gives the strain energy density function for the material model considering a similar shape setup. Here λ_a and λ_b represents the material constants. λ_1,λ_2 and λ_3 are the principal stretch ratios for the given two dimensional shape configuration. The strain energy is thus given as:

$$\Pi_m = \pi R^2 (\lambda_a (n^2 + (\frac{1}{n})^2 - 3) - 2\lambda_b) \tag{0.8}$$

0.5 The Equivalent Relationship

The final form of the equivalence between the material model and the shape model is given as

$$(\frac{\pi R^4}{4})^2(\delta_1(\frac{1}{n^2}-1)^2+\delta_2(n^2-1)^2)=\pi R^2(\lambda_a(n^2+(\frac{1}{n})^2-3)-2\lambda_b) \tag{0.9}$$

In order to solve this equation we consider a system of linear equations where we vary the value of ellipticity and solve for λ_a and λ_b using fixed values of δ_1 and δ_2
We construct a matrix of the form $\mathbf{Ax}=\mathbf{b}$

$$\begin{bmatrix}\pi R^2(n_1^2+\frac{1}{n_1^2}-3) & -2\\ \pi R^2(n_2^2+\frac{1}{n_2^2}-3) & -2\end{bmatrix}\begin{bmatrix}\lambda_a\\ \lambda_b\end{bmatrix}=\begin{bmatrix}\pi R^4 4)^2(\delta_1(\frac{1}{n_1^2}-1)^2+\delta_2(n_1^2-1)^2)\\ \pi R^4 4)^2(\delta_1(\frac{1}{n_2^2}-1)^2+\delta_2(n_2^2-1)^2)\end{bmatrix}$$

We find out these values of λ_a and λ_b and plot it against values of n and illustrate this result.

Friday, 29th June 2018

1 Equivalent Model Results using MATLAB

A piece of code is developed to compute the values of λ_a and λ_b. We create a new function and output the above two values using different values of the ellipticity factor.

```
function [y] = bestFit(delta, R, n1)
%n1 is the number of ellipticity factors, also used as iterators,
%used to solve the system of linear equations

a = delta(1);
b = delta(2);

d = (pi*(R^4)/4)^2;

counter = 1;

for i = 1:n1
for j = 1:n1
if (i ~= j)
A = [(i^2 + (1/i^2) - 3) -2; (j^2 + (1/j^2) - 3) -2];
B = [d*(a*((1/i^2)-1)^2 + b*(i^2 - 1)^2); d*(a*((1/j^2)-1)^2 + b*(j^2 - 1)^2)];
x = A\B;
t = [x ; i ;j];
y(counter,:) = t;
counter = counter + 1;
end
end
end
end
```

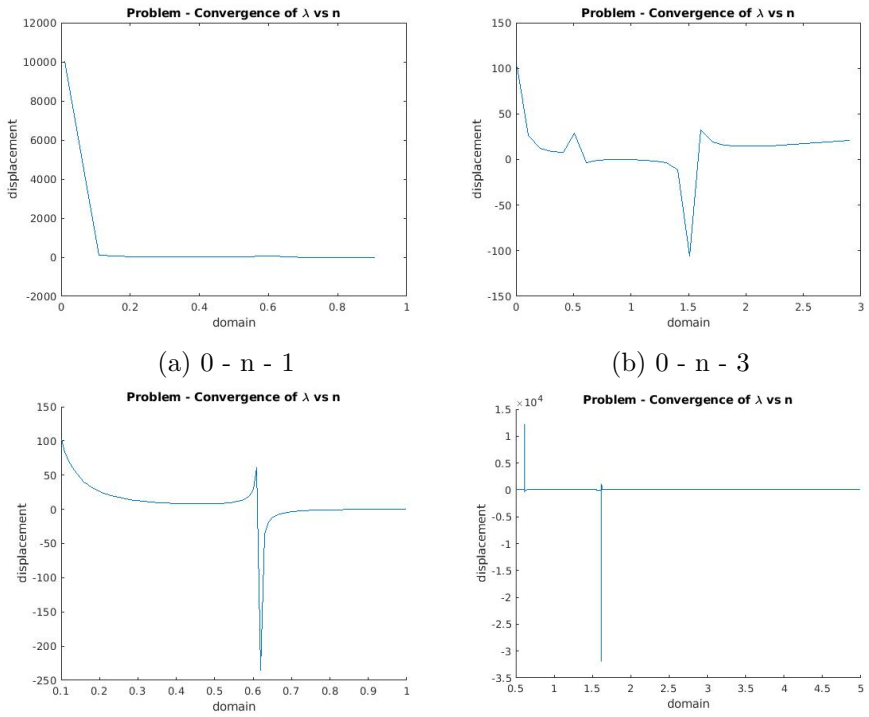
In order to graphically visualize the behaviour of λ with variation in ellipticity, we write another function considering same values of λ. **R** (Radius of the initial circular configuration) is kept as constant for the analysis. The user can input the values of δ_1 and δ_2 in this case.

```
clear all
close all
clc

delta = [1 1];
```

```
n = 1.2;
counter = 1;

for i = 1:0.001:n

lambda(counter) = (delta(1)*(1/(i^2) - 1)^2 + delta(2)*(i^2 - 1)^2)/(i^2 +
    (1/i^2) - 3);
%z = [lambda, i];
counter = counter + 1;

end

j = 1:0.001:n;
hold on
plot(j, lambda)
title('Problem - Convergence of \lambda vs n')
xlabel('domain')
ylabel('displacement')

%domain = ellipticity factor
%displacement = value of \lambda
```

Tuesday, 3 July 2018

1 Equivalent Model Results using MATLAB



(a) 0 - n - 1

(b) 0 - n - 3

(c) 0.1 - n - 1

(d) 0.5 - n - 5

Figure 1:
Variation of λ for varying ellipticity