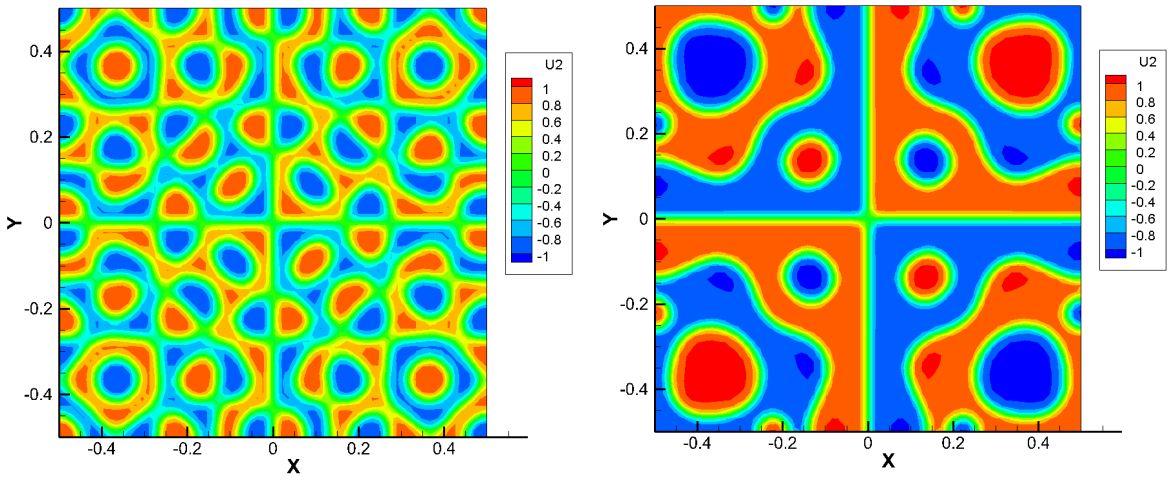
School of Engineering
Brown University

EN234: Computational methods in Structural and Solid Mechanics

Homework 8: Time dependent problems: The Cahn Hilliard Equation
Due Wed Nov 11, 2015



1. The Cahn-Hilliard equation is one of the most famous equations in materials science, and is an example of a more general description of materials with evolving structure known as 'phase field' models. The original purpose of the Cahn-Hilliard equation was to describe spinodal decomposition of a single-phase liquid or solid into two phases, but it has since been extended to model many other phenomena.

   The canonical Cahn-Hilliard equation describes a binary solution of A and B, whose composition is characterized by a variable c: $c=1$ corresponds to pure A, while $c=-1$ corresponds to pure B. The Gibbs free energy of the solution is often taken to be

$$F(c)=\frac{1}{4}(c^2-1)^2+\frac{1}{2}\kappa|\nabla c|^2$$

   where κ is a material property (quantifying the energy per unit area of an interface between A and B). The free energy has minima at $c=\pm 1$. It is therefore energetically favorable for the solid solution to phase separate into A- and B-rich regions. Moreover, it is energetically favorable for the system to minimize concentration gradients. As a result, a material that starts with c close to zero phase separates into regions of A and B, which then gradually coarsen, as shown in the figure.

   The Cahn-Hilliard equation describes how this process occurs. The concentration variable is governed by a diffusion equation

$$\frac{\partial c}{\partial t}=\nabla\cdot D\nabla\mu \qquad \mu=\frac{\delta G}{\delta c}=c(c^2-1)-\kappa\nabla^2 c$$

   where D is a diffusion coefficient (which we will assume is constant, but could depend on c).

Our goal is to solve this system of equations for *c(t)*, given some initial conditions. To keep things simple, we will consider a 2D rectangular region of material (shown in the figure), with symmetry boundary conditions (so $\nabla c \cdot \mathbf{n} = \nabla \mu \cdot \mathbf{n} = 0$ on all boundaries).

The first step is to set up a finite element approximation to the PDEs. This is usually done by solving simultaneously for μ and *c* (this avoids having to solve a fourth-order PDE). Introducing variations of $\delta\mu, \delta c$, you should be able to show that the weak form of the governing equations is

$$\int_V \frac{\partial c}{\partial t}\delta c dV - \int_V D\frac{\partial \mu}{\partial x_i}\frac{\partial \delta c}{\partial x_i}dV = 0 \qquad \int_V \mu\delta\mu dV - \int_V \left(c(c^2-1)\delta\mu + \kappa\frac{\partial c}{\partial x_i}\frac{\partial \delta\mu}{\partial x_i}\right)dV = 0 \qquad \forall\{\delta c, \delta\mu\}$$

Introducing finite element interpolation functions for μ and *c* in the usual way

$$\mu = N^a\mu^a \qquad c = N^a c^a \qquad\qquad \mu = N^b\delta\mu^b \qquad c = N^b\delta c^b$$

then yields the discrete system of equations

$$M_{ab}\frac{dc^b}{dt} + DK_{ab}\mu^b = 0 \qquad\qquad M_{ab}\mu^b - H^a(c^b) + \kappa P_{ab}c^b = 0$$

where

$$M_{ab} = \int_V N^a N^b dV \qquad P_{ab} = \int_V \frac{\partial N^a}{\partial x_i}\frac{N^b}{\partial x_i}dV \qquad H^b = \int_V c(c^2-1)N^b dV$$

Finally, we need a way to integrate this discrete system of equations with respect to time. As in all FEA problems, we use a time-marching scheme: given values μ^a, c^a at time *t*, we find the increments $\Delta\mu^a, \Delta c^a$ during the next time interval Δt , and then update the solution. A backward-Euler integration is used for μ

$$M_{ab}\left(\mu^b + \Delta\mu^b\right) - H^a(c^b + \Delta c^b) - \kappa P_{ab}(c^b + \Delta c^b) = 0$$

while a generalized mid-point time integration scheme is used to integrate the equation for the concentration

$$M_{ab}\frac{\Delta c^b}{\Delta t} + DP_{ab}\left[(1-\theta)\mu^b(t) + \theta(\mu^b(t) + \Delta\mu^b)\right] = 0$$

where $0 < \theta < 1$ is a numerical parameter. Choosing $\theta = 0$ gives a forward-Euler scheme; choosing $\theta = 1$ gives a backward-Euler scheme, and $\theta = 0.5$ (the usual choice) gives a mid-point scheme.

We now have a system of nonlinear equations to solve for $\Delta\mu^a, \Delta c^a$ and are on familiar territory.

To implement this idea, we need to re-write it as the usual set of finite element operations, as follows:

- Note that each node in the finite element mesh will have two degrees of freedom: the value of μ and the value of *c*. At a generic time-step, we will be solving for $\Delta\mu^a, \Delta c^a$ at each node.
- Instead of the usual B matrix that maps displacements to strains, we can introduce a modified B matrix that maps nodal values of μ^a, c^a to $\mu, c, \partial\mu/\partial x_i, \partial c/\partial x_i$. Thus

$$\begin{bmatrix} \mu \\ c \\ \partial\mu/\partial x_1 \\ \partial\mu/\partial x_2 \\ \partial c/\partial x_1 \\ \partial c/\partial x_2 \end{bmatrix} = [\mathbf{B}] \begin{bmatrix} \mu^1 \\ c^2 \\ \mu^2 \\ c^2 \\ \mu^3 \\ c^3 \\ \vdots \end{bmatrix}$$

$$[\mathbf{B}] = \begin{bmatrix} N^1 & 0 & N^2 & \\ 0 & N^1 & 0 & \\ \partial N^1/\partial x_1 & 0 & \partial N^2/\partial x_1 & \cdots\cdots \\ \partial N^1/\partial x_2 & 0 & \partial N^2/\partial x_2 & \\ 0 & \partial N^1/\partial x_1 & 0 & \\ 0 & \partial N^1/\partial x_2 & 0 & \end{bmatrix}$$

- The finite element stiffness matrix and residual vector can then be expressed as

$$[k^{el}] = \int_V [\mathbf{B}]^T [\mathbf{D}][\mathbf{B}] dV \qquad \underline{r^{el}} = -\int_V [\mathbf{B}]^T \underline{q} dV$$

Where

$$\underline{q} = \begin{bmatrix} \mu + \Delta\mu - f(c+\Delta c) \\ \Delta c/\Delta t \\ -\kappa\partial(c+\Delta c)/\partial x_1 \\ -\kappa\partial(c+\Delta c)/\partial x_2 \\ D\partial(\mu+\theta\Delta\mu)/\partial x_1 \\ D\partial(\mu+\theta\Delta\mu)/\partial x_2 \end{bmatrix} \qquad f(c) = c(c^2-1)$$

$$[\mathbf{D}] = \begin{bmatrix} 1 & -df/dc & 0 & 0 & 0 & 0 \\ 0 & 1/\Delta t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\kappa & 0 \\ 0 & 0 & 0 & 0 & 0 & -\kappa \\ 0 & 0 & \theta D & 0 & 0 & 0 \\ 0 & 0 & 0 & \theta D & 0 & 0 \end{bmatrix} \qquad df/dc = 3(c+\Delta c)^2 - 1$$

Your goal is to implement this in EN234FEA. Some guidelines:

- Your nodes have 2 coordinates and 2 DOF; you can use the same variables to store these as in your 2D linear elasticity codes.
- You can use any of the standard family of 2D elements. They have 3 properties: the diffusion coefficient D, κ , and θ
- You will find that you can solve the problem with rather minor changes to the 2D elasticity code you wrote for Homework 3 – you merely need to re-define the **B** matrix and **D** matrix

(note that **D** depends on concentration, and so must be evaluated inside the integration loop), and implement a procedure to calculate the vector $\underline{q}$.

- You don't need to project values from integration points to the nodes, so you can just delete the field projection subroutine for this element.
- There are no direct or forced boundary conditions in this problem
- You will need to define an initial value for the concentration. The solution shown in the figure was generated with

$$c(t=0) = 0.01\sin(15x_1)\sin(15x_2)$$

  EN234FEA has a user-subroutine that can be used to define initial values of degrees of freedom, and this function has already been coded for you. Feel free to change it – the pattern you get is determined by the initial conditions.
- As an example, run solutions for a square region $-0.5 < x_1 < 0.5 \quad -0.5 < x_2 < 0.5$, with parameter values as follows:
  - $D = 1,\ \kappa = 0.0001,\ \theta = 0.5$
  - Time step 0.001

  You can run 50 steps or so (the evolution slows down towards the end so running longer simulations gets boring).
- To save you some time, an input file setting up this problem has been provided for you in a file called Cahn_hilliard_2d.in. Edit the file to run the code with CHECK STIFFNESS before trying to run a full simulation, and try it with just a few steps before running all 50 steps. The full simulations might take a minute or two to run in Debug mode.