Phase-Field Modeling with Finite Element Method

Xiukun Hu[1,2], *Mallory Lai*[1,3], *Geeta Monpara*[1,4]

11/11/2016

[1] University of Wyoming, [2] Department of Mathematics, [3] Department of Botany [4] Department of Mechanical Engineering

Since the underlying microstructure of a material can affect its physical properties, it is important to understand the processes underlying microstructure formation in order to improve engineered materials.[1] However, modeling microstructure formation is complicated by the fact that the structures are thermodynamically unstable and thus evolve over time.[2] Phase-field modeling has gained popularity as a technique for simulating microstructure evolution due to its ability to incorporate different thermodynamic driving forces into the model. As such, it can be used to simulate the microstructure evolution seen in such processes as solidification, grain growth, and solid-state phase transformations.[2]

When it comes to phase-field modeling, the finite element method (FEM) underlies some of the most efficient and accuracte numerical methods to date.[3] The strength of the finite element method arises from its ability to model irregular shapes, heterogeneous materials, general load conditions, nonlinear behavior, and dynamic effects. Furthermore, once the finite element model is established, it can be modified relatively easily.[4] Finite element analysis involves the conversion of the original partial differential equation (PDE) from its original, strong form, into a weak integral form. The domain is then discretized into elements from which shape functions are used to find an approximate numerical solution.[5]

We present here the Galerkin finite element in 2D. The Galerkin method is an example of a weighted residual method which assumes the same shape functions are used for both the solution and the weights.[5] We begin by identifying the strong form PDE and rearranging to get zero on the right hand side. We then multiply the resulting equation by our shape function, as per the Galerkin method.[6] Finally, we integrate over the domain, accounting for both local and global coordinate systems.

Phase field equation: In the current work, we model solidification of a one component system with the phase field method. We solve phase field equations by finite element analysis. Our phase field variable ϕ assumes values between 0 and 1, where value 0 represents the liquid state and 1 represents solid state. We began with the Ginzburg-Landau entropy functional $S(\phi)$.[1] Using variational calculus, we further minimize this functional with respect to ϕ and arrive at the following partial differential equation:

$$\frac{\partial \phi}{\partial t} = 2\left(\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2}\right) - g(\phi)$$

where,

$$g(\phi) = a\phi^3 + b\phi^2 + c\phi$$

with

$$a = \frac{-36}{\epsilon^2}$$

$$b = \frac{54}{\epsilon^2} + \frac{6m}{\epsilon}$$

and

$$c = -\left(\frac{18}{\epsilon^2} + \frac{6m}{\epsilon}\right)$$

where, ϵ determines how diffused the solid-liquid interface would be while m is a driving force for the interface.

This differential equation represents the evolution of a phase field variable ϕ in time with the control variables ϵ and m.

$$\Rightarrow \frac{\partial \phi}{\partial t} - 2\nabla \cdot \nabla\phi + g(\phi) = 0$$

Applying the Galerkin method:

$$\int\int [N]^T \{\frac{\partial \phi}{\partial t} - 2\nabla \cdot \nabla\phi + g(\phi)\} dxdy = 0$$

Working in a local coordinate system $(\xi - \eta)$ and distributing:

$$\int_{\Omega_{e_i}} \int \left([N]^T \frac{\partial \phi}{\partial t} - [N]^T 2\nabla \cdot \nabla\phi + [N]^T g(\phi)\right) dxdy$$

First term:

$$\int_{\Omega_{e_i}} \int \left([N]^T \frac{\partial \phi}{\partial t}\right) dxdy$$

Replacing global coordinates with local (elemental) coordinates $(\xi - \eta)$ for a rectangualr element:

$$= \int_{-1}^{1}\int_{-1}^{1} \left([N(\xi,\eta)]^T \frac{\partial \phi(\xi,\eta)}{\partial t}|J|\right) d\xi d\eta$$

$$= \int_{-1}^{1}\int_{-1}^{1} \left([N(\xi,\eta)]^T \frac{\partial([N(\xi,\eta)][\phi_{e_i}]^T)}{\partial t}|J|\right) d\xi d\eta$$

$$= |J| \int_{-1}^{1}\int_{-1}^{1} \left([N]^T[N][\dot{\phi}_{e_i}]^T)\right) d\xi d\eta$$

$$= \left[|J| \int_{-1}^{1}\int_{-1}^{1} \left([N]^T[N])\right) d\xi d\eta\right] [\dot{\phi}_{e_i}]^T$$

where

$$[\dot{\phi}_{e_i}]^T = \frac{\partial [\phi_{e_i}])^T}{\partial t}$$

$$= \frac{\left[\phi_{e_i}^{n+1}\right]^T - \left[\phi_{e_i}^{n}\right]^T}{\Delta t}$$

First term =

$$[C_{e_i}][\dot{\phi}_{e_i}]^T \tag{i}$$

Second term:

$$\int_{\Omega_{e_i}} \int \left([N]^T 2\nabla \cdot \nabla\phi\right) dxdy$$

Applying Green's theorem/Divergence theorem:

$$= 2\{-\int_{\Omega_{e_i}}\int \nabla\phi \cdot [N]^T dxdy + \oint N_i \nabla\phi\eta dl\}$$

$$= (-[K]_{ei} + [BC_x]_{ei} + [BC_y]_{ei})[\phi_{ei}]^T$$

Where,

$$[K]_{ei} = 2\int_{\Omega_{e_i}}\int [B(1,:)]^T[B(1,:) + [B(1,:)]^T[B(1,:)]|J|d\xi d\eta$$

$$[BC_x]_{ei} = 2\int_S [N]^T[B(1,:)]dx$$

$$[BC_x]_{ei} = 2\int_S [N]^T[B(2,:)]dy$$

and

$$[B] = \begin{bmatrix} N_{1,x} & N_{2,x} & N_{3,x} & N_{4,x} \\ N_{1,y} & N_{2,y} & N_{3,y} & N_{4,y} \end{bmatrix}$$

Third term:

$$\int_{\Omega_{e_i}}\int \left([N]^T g(\phi)\right) dxdy$$

$$I_1 = \int_{\Omega_{e_i}}\int \left([N]^T \phi\right) dxdy$$

$$I_1 = \int\int \left([N]^T[N][\phi_{e_i}]^T\right) dxdy$$

$$= \int_{-1}^{1}\int_{-1}^{1} \left([N]^T[N][\phi_{e_i}]^T|J|\right) d\xi d\eta$$

$$\Rightarrow \phi = [N][\phi_{e_i}]^T \Rightarrow [N]^T\phi = [C_{e_i}[\phi_{e_i}]]$$

$$\Rightarrow \phi^2 = \left([N][\phi_{e_i}]^T\right)^2$$

$$= [\phi_{e_i}][N]^T[N][\phi_{e_i}]^T$$

$$= [\phi_{e_i}][C_e][\phi_{e_i}]^T$$

where:

$$[C_e] = [N]^T[N]$$

$$\Rightarrow \phi^3 = \left([N][\phi_{e_i}]^T\right)^3$$

$$= [N][\phi_{e_i}]^T[\phi_{e_i}][N]^T[N][\phi_{e_i}]^T$$

$$= [N][\phi_{e_i}]^T[\phi_{e_i}][C_e][\phi_{e_i}]^T$$

$$= [\phi_{e_i}]^T[\phi_{e_i}]\underbrace{[N][C_{e_i}]}_{[D_{e_i}]}[\phi_{e_i}]^T$$

$$= [\phi_{e_i}]^T[\phi_{e_i}][D_{e_i}][\phi_{e_i}]^T$$

$$[N]^T\phi^3 = [\phi_{e_i}]^T[\phi_{e_i}][C_{e_i}]^2[\phi_{e_i}]^T$$

Hence,

$$\int_{\Omega_{e_i}} \int \left([N]^T g(\phi)\right) dxdy$$

$$= \int_{-1}^{1}\int_{-1}^{1} \left([N]^T(a\phi^3 + b\phi^2 + c\phi)|J|\right) d\xi d\eta$$

$$= \left[a\int_{-1}^{1}\int_{-1}^{1} \left([\phi_{e_i}]^T[\phi_{e_i}][C_{e_i}]^2|J|\right) d\xi d\eta + b\int_{-1}^{1}\int_{-1}^{1} \left([N]^T[\phi_{e_i}][C_{e_i}]|J|\right) d\xi d\eta + \underbrace{c\int_{-1}^{1}\int_{-1}^{1} \left([C_{e_i}]\right) d\xi d\eta}_{G_e} \right] [\phi_{e_i}]^T$$

Writing phase-field equation in terms of matrices:

$$= |J| \left[\int_{-1}^{1}\int_{-1}^{1} [C_e] d\xi d\eta\right] [\dot{\phi}_{e_i}]^T$$

$$= 2[K_e][\phi_{e_i}]^T - [G_e][\phi_{e_i}]^T$$

Where,

$$[C_{e_i}] = [N]^T[N]$$

Methods

In order to implement the FEM, we began by considering the equation without $g(\phi)$. By witholding $g(\phi)$, we can then compare our results to the heat equation. Be began with a triangular mesh generating code written in Matlab by John Burkardt.[7] From here, we modified the code to fit our model. We started with very simple initial conditions, letting a square within our rectangular domain represent a solid phase with $\phi = 1$ and elsewhere a liquid phase with $\phi = 0$ (Fig.1). Neumann, Dirichlet, and Periodic boundary conditions were imposed.

After successful implementation of all three boundary conditions without $g(\phi)$, we added $g(\phi)$ to our modified code. Instead of a square domain, we produced a rectangular strip. We varied ϵ to see if changes in the diffusion of the interface could be detected. Then we varied m to see if changes in direction and speed at the interface could be seen. We compared our results with those seen in Nestler[1].

Results and Discussion

Without $g(\phi)$:

Modification of our code without $g(\phi)$ produced results aligning with those of the heat equation.

(1) Neumann Boundary Conditions (Fig. 2):

$$\nabla\phi \cdot_{\partial\Omega} = 0$$

Implementing for a rectangular box:

$$\phi_{0,j} = \phi_{1,j}; \phi_{N_x,j} = \phi_{N_x-1,j}$$

$$\phi_{i,0} = \phi_{i,1}; \phi_{i,N_y} = \phi_{i,N_y-1}$$

where, $i/j = 0, 1, 2, ...N_x/N_y$

(2) Dirichlet Boundary Conditions (Fig. 3): Here, all $\phi_{i,j} = 0$ at all the boundary points

(3) Periodic Boundary Conditions (Fig. 4):

$$\phi_{0,j} = \phi_{N_x-1,j}; \phi_{N_x,j} = \phi_{1,j}$$

$$\phi_{i,0} = \phi_{i,N_y-1}; \phi_{i,N_y} = \phi_{i,1}$$

where, $i/j = 0, 1, 2, ...N_x/N_y$

With $g(\phi)$:

We further modified our code to implement $g(\phi)$. To benchmark our results with Nestler[1], we simulated the following cases:

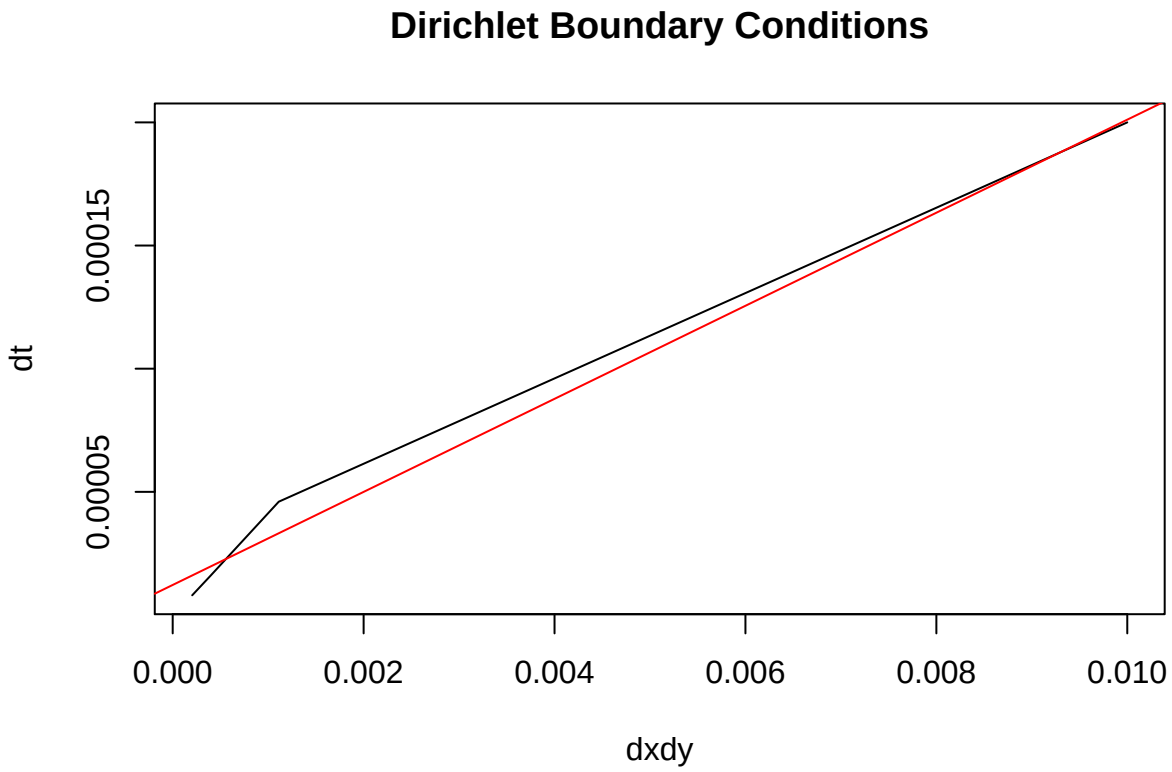
(1) $\epsilon = 1$ and 10 while keeping m = 0: These conditions imply that as the driving force is zero, interface should not move. However, two different values of ϵ show the difference in interfacial thickness. Animations are contained in the files named fem_eps_1m_0.mp4 and fem_eps_10m_0.mp4 in the Dropbox folder.

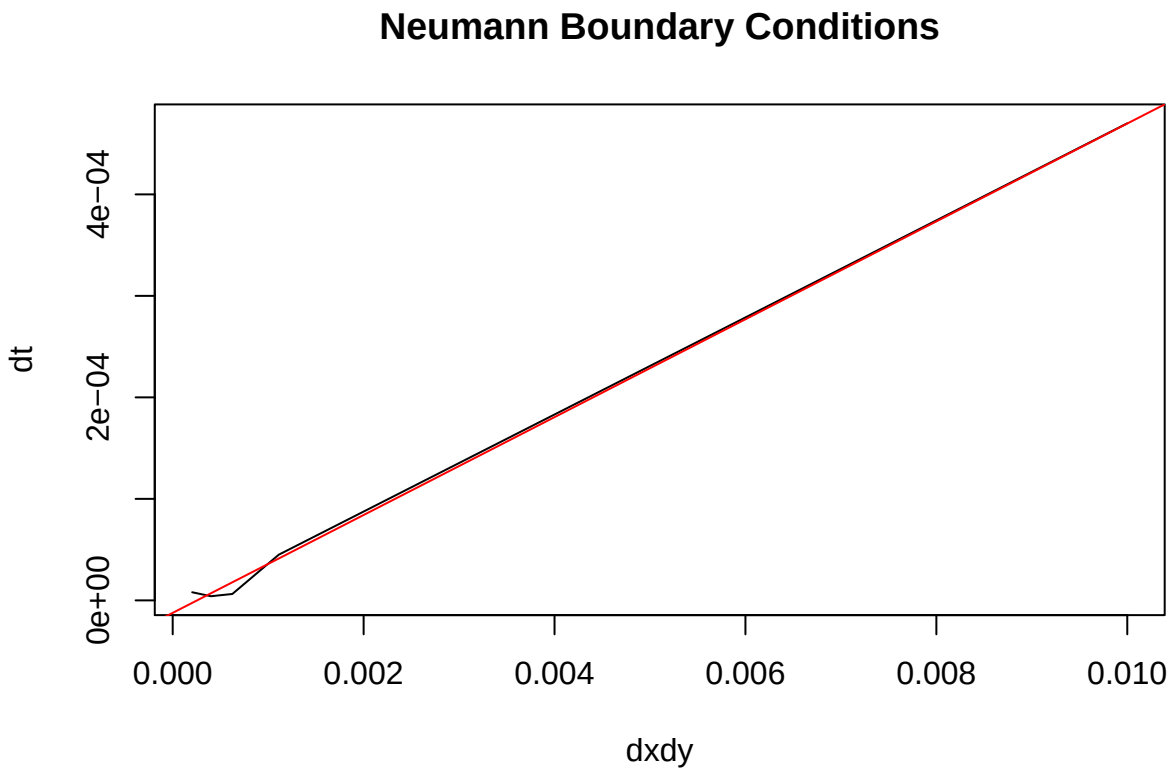
(2) $m = 1$ and -1 while keeping $\epsilon = 1$: These conditions should make the interface move in the opposite directions with respect to each other. For $m = 1$, the interface will move downwards. In other words, $m = 1$ will make the solid phase ($\phi = 1$) grow, resulting in solidification, while $m = -1$ will result in liquification where the liquid phase will consume solid phase. These results compare excellently with Nestler.[1] Again, the animations are contained in the files named as fem_eps_1m_1.mp4 and fem_eps_1m_-1.mp4 in the Dropbox folder.

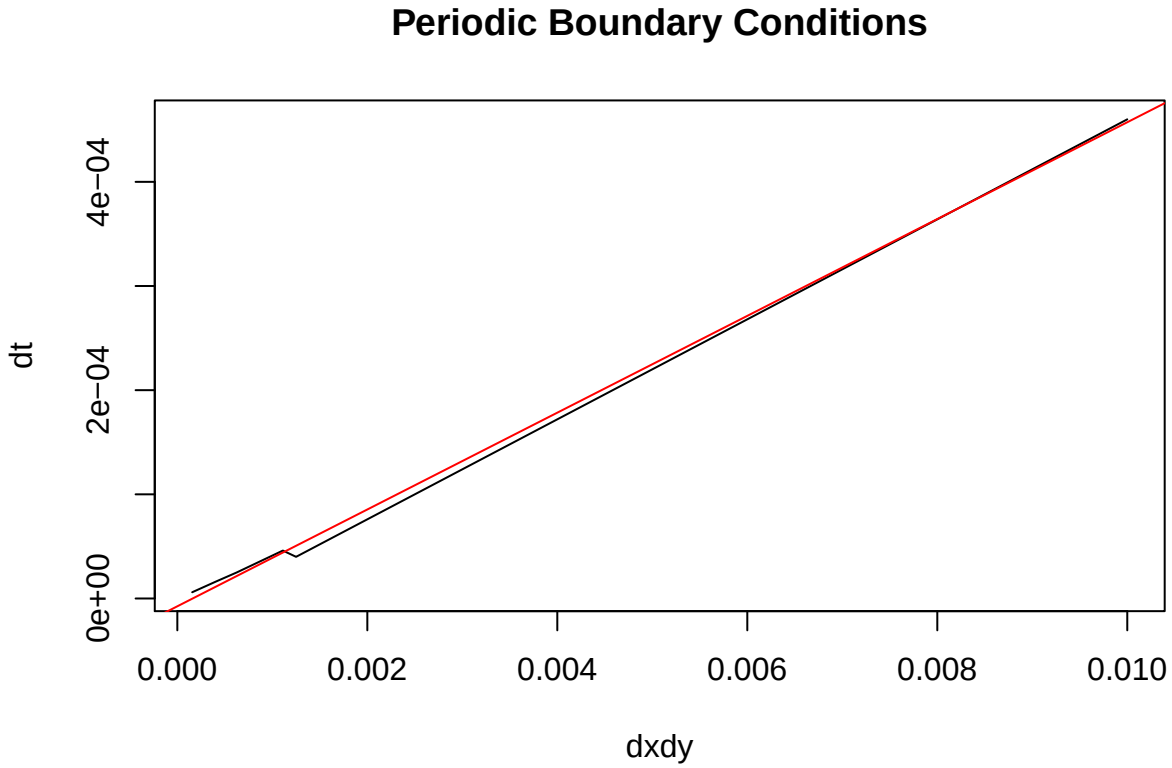
(3) Simulating Random Microstructure: Here, we populated the liquid melt ($\phi = 0$ matrix) with multiple solid phase seeds (zones of $\phi = 0$). Periodic boundary conditions were applied to the simulation box. Values of m and ϵ are 5 and 1, respectively. Animation of this simulation is in the Dropbox folder as random_microstructure.mp4. As seen from this animation, we find that the problem behaves as heat diffusion with an increasing amount liquid, which eventually converts to solid ($\phi = 1$). This situation seems unrealistic as our differential equation does not account for temperature effect. Hence, with control parameter m = 5, solid phase should strictly grow, while we do not observe the same.

For the stability analysis, we changed the discretization to see its effects on stability. We get a maximum value for dt (dt_max) given a mesh size. For dt values less than dt_max, the analysis will be stable. In the figures, we show the variations of dt_max with dx*dy, where dx and dy is the mesh size in the x and y directions. We find a linear relationship between the two variables. This behavior is consistent among the

different boundary conditions used. However, it changes upon manipulation of the size of the domain and initial conditions.







Insert error analysis stuff

Further analysis and comparison with other methods are required to resolve the problems surrounding the simulation of random microstructure. Future work should center around speeding up the current code. Converting the code from Matlab to C, and using techniques for sparse matrices, we believe the code could be sped up considerably.

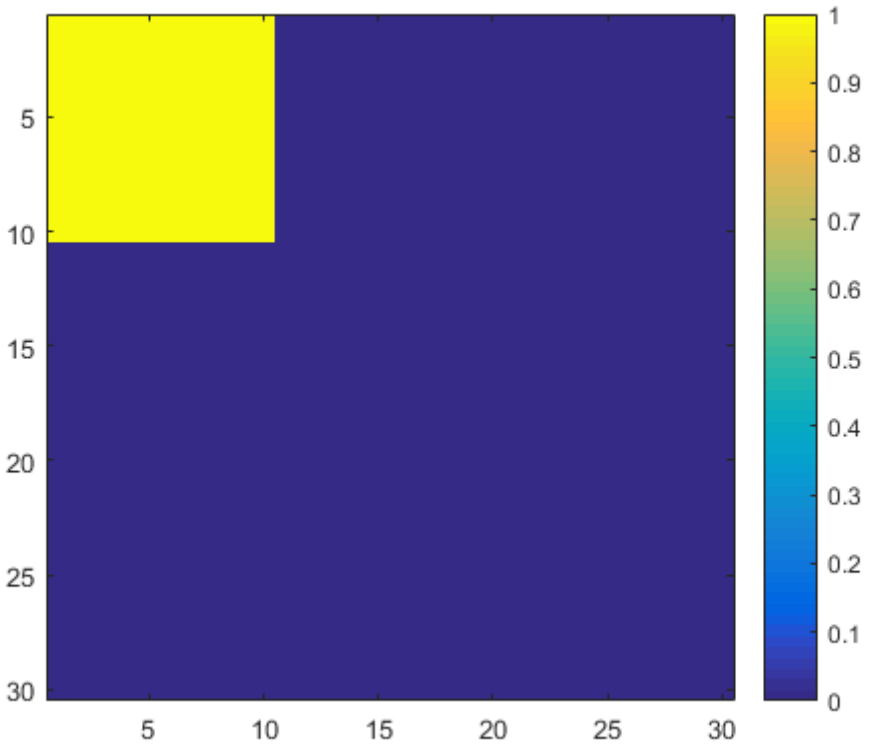


Figure 1: Initial conditions for our PFM. The rectangular region represents $\phi = 1$ and elsewhere $\phi = 0$.

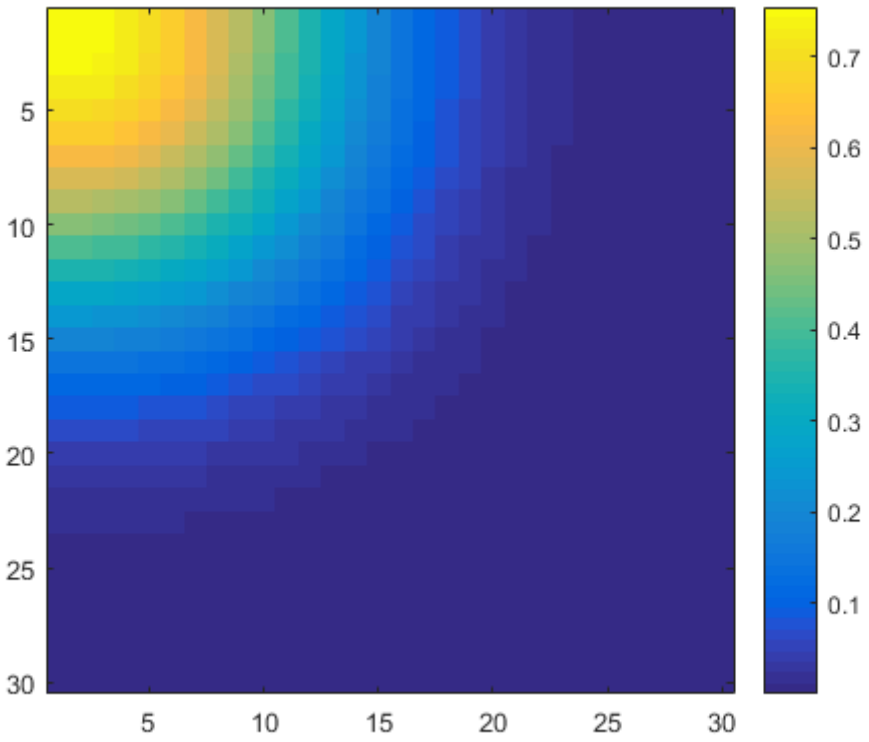


Figure 2: Neumann boundary conditions for our PFM without $g(\phi)$.

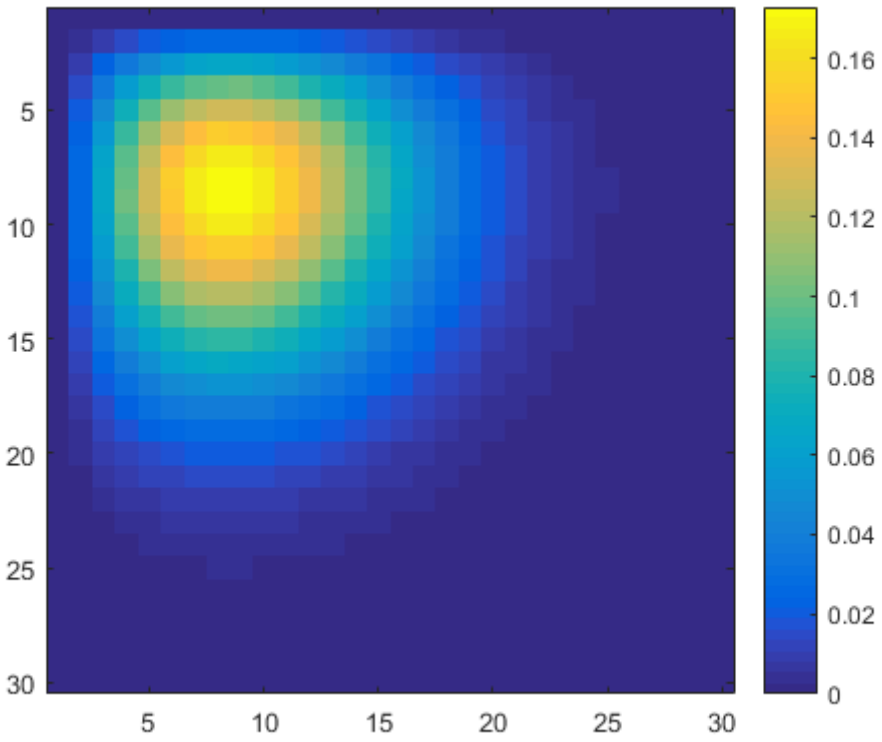


Figure 3: Dirichlet boundary conditions for our PFM without $g(\phi)$.

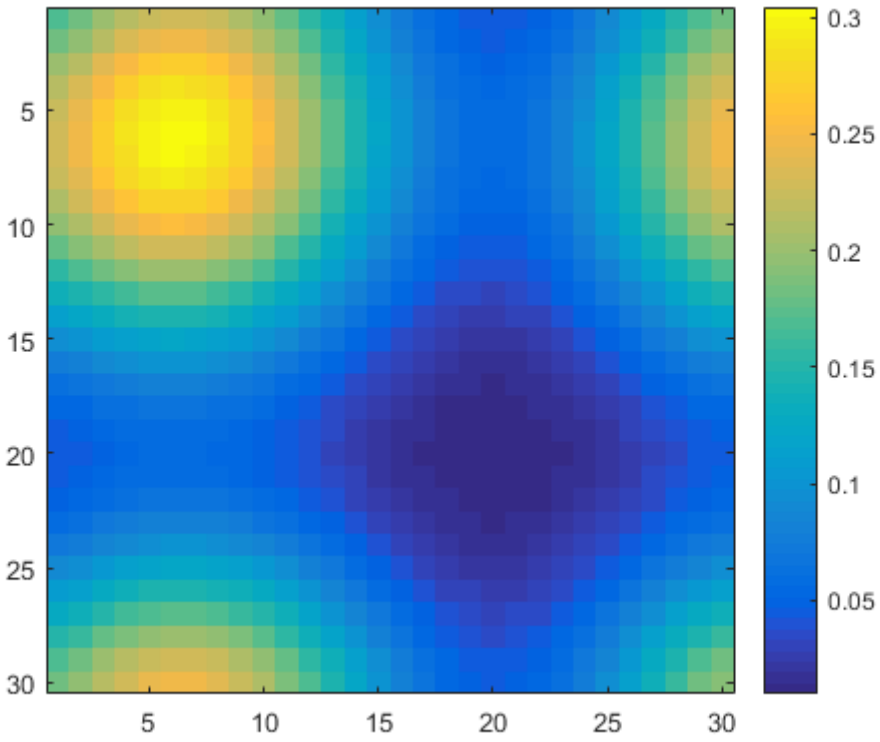


Figure 4: Periodic boundary conditions for our PFM without $g(\phi)$.

Supplementary materials

Found in Dropbox folder ma5340f16 within the LMH-phase-field-model folder.

References

1. Nestler, B. 7 - Phase-Field Modeling, In Computational Materials Engineering, Academic Press, Burlington, 2007, Pages 219-266, http://dx.doi.org/10.1016/B978-012369468-3/50007-1.
2. Moelans, N., Blanpain, B., & Wollants, P. (2008). Calphad: An introduction to phase-field modeling of microstructure evolution Pergamon Press. doi:10.1016/j.calphad.2007.11.003
3. Provatas, N., Elder, K. Phase-Field Methods in Material Science and Engineering. Wiley-VCH, 2010.
4. Logan, D. A First Course in the Finite Element Method, Fourth Edition. Thomson, 2007.
5. Becker, T., & Kaus, B. Numerical Modeling of Earth Systems: An introduction to computational methods with focus on solid Earth applications of continuum mechanics. USC, Los Angeles, 2016.
6. MOOSE Framework. Finite Elements:The MOOSE way. http://mooseframework.org/wiki/MooseTraining/FEM/
7. Burkardt, J. fem2d_poisson_rectangle_linear.m 2010.