This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier’s archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/copyright
A quantitative comparison between $C^0$ and $C^1$ elements for solving the Cahn–Hilliard equation

Liangzhe Zhang$^{a,b}$, Michael R. Tonks$^{a,*}$, Derek Gaston$^a$, John W. Peterson$^a$, David Andrs$^a$, Paul C. Millett$^a$, Bulent S. Binera$^a$

$^a$Fuels Modeling and Simulation Department, Idaho National Laboratory, Idaho Falls, ID 83415, United States

$^b$Stress Engineering Services Inc., Houston, TX 77041, United States

ARTICLE INFO

Article history:
Received 5 December 2011
Received in revised form 1 December 2012
Accepted 3 December 2012
Available online 13 December 2012

Keywords:
Cahn–Hilliard equation
FEM
Accuracy
Computational time
JFNK

ABSTRACT

The Cahn–Hilliard (CH) equation is a time-dependent fourth-order partial differential equation (PDE). When solving the CH equation via the finite element method (FEM), the domain is discretized by $C^1$-continuous basis functions or the equation is split into a pair of second-order PDEs, and discretized via $C^0$-continuous basis functions. In the current work, a quantitative comparison between $C^1$ Hermite and $C^0$ Lagrange elements is carried out using a continuous Galerkin FEM formulation. The different discretizations are evaluated using the method of manufactured solutions solved with Newton’s method and Jacobian-Free Newton Krylov. It is found that the use of linear Lagrange elements provides the fastest computation time for a given number of elements, while the use of cubic Hermite elements provides the lowest error. The results offer a set of benchmarks to consider when choosing basis functions to solve the CH equation. In addition, an example of microstructure evolution demonstrates the different types of elements for a traditional phase-field model.

Published by Elsevier Inc.

1. Introduction

The Cahn–Hilliard (CH) equation was originally introduced to describe spinodal decomposition, wherein components of a binary system spontaneously partition into two pure phases [1,2]. Since then, the CH equation has been widely used to model material processes over a diverse range, one prominent example being the phase field method [3]. The importance of the CH equation has inspired a series of theoretical analyses to guarantee satisfactory solutions [4–7], and the CH equation has been solved with a variety of numerical algorithms, including finite difference methods [8–10], finite element methods (FEM) [11–19], finite volume methods [20], spectral methods [21,22], and others [23].

In the current study, we focus on solving the CH equation within the framework of a continuous Galerkin FEM, and several implementations are possible within this context. For example, cubic Hermite basis functions have been applied to directly solve the weak form of the fourth-order CH equation, enforcing $C^1$ continuity of the numerical solution [12,14]. Comparatively more effort has been devoted to the approximation of the CH equation using $C^0$ finite elements [24,13,25]. This stems from the fact that the fourth-order CH equation can be split into two second-order PDEs by introducing an auxiliary variable [24], and $C^0$ basis functions are therefore applicable. However, a detailed quantitative comparison of the computational time and accuracy of the different approaches has never been carried out previously, and this motivates the current investigation.

* Corresponding author. Tel.: +1 208 526 6319.
E-mail address: michael.tonks@inl.gov (M.R. Tonks).

0021-9991/$ – see front matter Published by Elsevier Inc.
http://dx.doi.org/10.1016/j.jcp.2012.12.001
The numerical examples in this paper were computed with MARMOT [26], a phase-field (PF) code based on the Multiphysics Object-Oriented Simulation Environment (MOOSE) [27,28] library developed at the Idaho National Laboratory. MOOSE solves the nonlinear systems of equations associated with the FEM via either the (inexact) Newton method or the Jacobian-Free Newton Krylov (JFNK) method [29]. In the present study, the CH equation is solved with $C^1$ and $C^0$ basis functions for several representative test cases. The computational time and error (in both $L_2$ and $H^1$ norms) are recorded. The subsequent comparisons show that linear Lagrange elements dramatically reduce computational time, while the cubic Hermite elements offer a low computational error. More detail is given in the following sections.

2. Weak forms of the Cahn–Hilliard equation

The fourth-order CH equation is written as

$$\frac{\partial c}{\partial t} = \nabla \cdot \left( M \nabla \left( f'(c) - \epsilon \nabla^2 c \right) \right), \tag{1}$$

where $c$ is a conservative order parameter, $M$ is the mobility of $c$, $\epsilon$ is a constant, and $f(c)$ is a free energy density function. To solve the CH equation using FEM, Eq. (1) is multiplied by a test function $\phi \in H^2$ ($H^2$ is the Hilbert space of functions with square-integrable second derivatives) and integrated by parts twice [14,30]. We then seek $c \in H^2$ and satisfying the boundary conditions, such that

$$\left( \frac{\partial c}{\partial t}, \phi \right) = -\epsilon \left( \nabla^2 c, \nabla \cdot (M \nabla \phi) \right) - (M \nabla f'(c), \nabla \phi) + \left( M \nabla \left( f'(c) - \epsilon \nabla^2 c \right) \cdot \bar{n}, \phi \right) + \left( \epsilon \nabla^2 c, M \nabla \phi \cdot \bar{n} \right), \tag{2}$$

for all admissible $\phi$, where $(\cdot, \cdot)$ and $(\cdot, \cdot)$ stand for interior and boundary inner products, respectively.

On the other hand, Eq. (1) can be split into two second-order PDEs by introducing an auxiliary variable $\mu$ [24], which leads to

$$\frac{\partial c}{\partial t} = \nabla \cdot (M \nabla \mu), \tag{3}$$

$$\mu = f'(c) - \epsilon \nabla^2 c,$$

where $\mu$ is known as the chemical potential [24]. The weak statement in this case is: find $(c, \mu) \in H^1 \times H^1$ and satisfying the boundary conditions, such that

$$\left( \frac{\partial c}{\partial t}, \phi \right) = -\left( M \nabla \mu, \nabla \phi \right) + \left( M \nabla \mu \cdot \bar{n}, \phi \right), \tag{4}$$

$$\left( \mu, \psi \right) = \left( f'(c), \psi \right) + \epsilon \left( \nabla c, \nabla \psi \right) - \epsilon \left( \nabla c \cdot \bar{n}, \psi \right), \tag{5}$$

for all admissible $(\phi, \psi)$. Although both discretizations (2) and (4)–(5) are capable of delivering satisfactory results [24,13,12,14], no detailed quantitative comparison between the two solution schemes has so far been conducted. The current research employs cubic Hermite and (linear and quadratic) Lagrange finite elements to solve (2) and (4)–(5), respectively. We note that the use of cubic Hermite elements on non-rectangular domains is possible [31], however we do not pursue the isoparametric formulation in the present work. For simplicity we set $M$ and $\epsilon$ to unity, and assume the following form for the free energy density function:

$$f(c) = \frac{c^4}{4} - \frac{c^2}{2}. \tag{6}$$

The discretization in time is via the Crank–Nicolson method [32], and the Newton and JFNK solvers are based on PETSc’s [33] “SNES” module. Both discretizations are solved via the generalized minimal residual (GMRES) method with ILU (1). Each nonlinear iteration is solved to a relative tolerance of $10^{-10}$, while the linear iterations within each nonlinear iteration are solved to a relative tolerance of $10^{-4}$. Three specific finite element discretizations are considered:

1. A $C^1$ Hermite discretization of Eq. (2). In this approach, the domain is discretized with four-node quadrilateral elements (Q4) in 2D and eight-node hexahedral elements (H8) in 3D.
2. A $C^0$ quadratic Lagrange discretization of Eqs. (4) and (5) employing nine-node quadrilateral elements (Q9) in 2D and 27-node hexahedral elements (H27) in 3D.
3. A $C^0$ linear Lagrange discretization of Eqs. (4) and (5) employing Q4 elements in 2D and H8 elements in 3D.

To facilitate comparison between the three approaches, information regarding the number of degrees of freedom (DOFs), quadrature points (QPs) and Jacobian matrix average bandwidth for each is listed in Table 1. Note that the second-order Lagrange elements have the most degrees of freedom for a given number of elements, while the third-order Hermite elements require the most quadrature points. In addition, the average bandwidth of the Jacobian matrix varies significantly between the three approaches. In 2D, the Hermite and the second-order Lagrange elements have nearly the same bandwidth, which is twice that of the first-order Lagrange elements. In 3D, the Hermite elements have the largest bandwidth by a factor of about 1.7.
Table 1
Spatial dimension, geometric element type, global degree-of-freedom count, global quadrature point count, and average matrix bandwidth for the Lagrange and cubic Hermite discretizations in a regular grid with N elements in each direction. The leading factor of 2 appearing in the DOF counts for the Lagrange discretizations accounts for the fact that we introduce an additional variable for this element type. The average bandwidth for the quadratic Lagrange elements is based on the fine mesh limit. In 2D, this means 50 non-zeros per “vertex” DOF, 36 non-zeros per “edge” DOF, and 18 non-zeros per “center” DOF, occurring in a 1:2:1 ratio. In 3D, it is equivalent to 250 non-zeros per “vertex” DOF, 150 non-zeros per “edge” DOF, 90 non-zeros per “face” DOF, and 54 non-zeros per “center” DOF, occurring in a 1:3:3:1 ratio.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Element</th>
<th>DOFs</th>
<th>QPs</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st O. Lagrange</td>
<td>2D</td>
<td>Q4</td>
<td>2(N + 1)^2</td>
<td>4N^2</td>
</tr>
<tr>
<td>3D</td>
<td>H8</td>
<td>2(N + 1)^3</td>
<td>8N^3</td>
<td>54</td>
</tr>
<tr>
<td>2nd O. Lagrange</td>
<td>2D</td>
<td>Q9</td>
<td>2(2N + 1)^2</td>
<td>9N^2</td>
</tr>
<tr>
<td>3D</td>
<td>H27</td>
<td>2(2N + 1)^3</td>
<td>27N^3</td>
<td>128</td>
</tr>
<tr>
<td>3rd O. Hermite</td>
<td>2D</td>
<td>Q4</td>
<td>4(N + 1)^2</td>
<td>16N^2</td>
</tr>
<tr>
<td>3D</td>
<td>H8</td>
<td>8(N + 1)^3</td>
<td>64N^3</td>
<td>216</td>
</tr>
</tbody>
</table>

3. Results

To evaluate the three solution approaches discussed above, we first conduct a detailed investigation using the method of manufactured solutions (MMS) [34] (Section 3.1). We then compare the results of a typical phase field simulation in which two circles coalesce over time (Section 3.2).

3.1. Manufactured solution comparison

The method of manufactured solutions proceeds in this particular case by assuming a “solution” \( \hat{c} \) of the form

\[
\hat{c} = (t + 1) \sin(\pi x) \tag{7}
\]

and then computing (by substituting \( \hat{c} \) into (1)) the “forcing function” \( \mathcal{F} \) which satisfies

\[
\frac{\partial \hat{c}}{\partial t} - \nabla \cdot [\nabla (f(\hat{c}) - \nabla^2 \hat{c})] = \mathcal{F}. \tag{8}
\]

For the particular \( \hat{c} \) given in (7), we obtain

\[
\mathcal{F} = \sin(\pi x) + 3(t + 1)^3 \pi^2 \sin^3(\pi x) + (t + 1) \pi^4 \sin(\pi x) - (t + 1) \pi^2 \sin(\pi x) - 6(t + 1)^3 \pi^2 \sin(\pi x)
\]

\[
\times \cos^2(\pi x). \tag{9}
\]

The modified Eq. (8) is then solved numerically on a unit square domain in 2D (unit cube in 3D) with a uniform grid of \( N^{\text{dim}} \) elements, for each of the finite element discretizations discussed in the preceding section. As \( N \) is increased, the computation time and the error (in the \( L_2 \) and \( H^1 \) norms) are recorded for each simulation. The computation time reported is the total wall time for one time step, not including the startup time. In addition, we solve both the 2D and 3D simulations using Newton’s method and JFNK and compare their performance. Since our focus in this study is on the spatial discretization, only one time step of size \( \Delta t = 1.0 \) is taken.

The theoretical error convergence rates [35] of the three types of elements are confirmed for both \( L_2 \) and \( H^1 \) -error using the smooth solution of Eq. (7). The JFNK solution delivered identical error to the Newton solve, and the corresponding 3D studies delivered the same convergence rates. In addition, we observe that the cubic Hermite elements provide superior accuracy over the two Lagrange elements for all mesh resolutions under consideration.

While the Hermite elements provide superior accuracy over the two Lagrange elements, they are also the most computationally expensive (see Fig. 2) and the linear Lagrange elements are the least. When using Newton’s method, the three approaches have a similar slope on a log–log plot of the computation time vs. the number of elements both in 2D (Fig. 2 and 3D (Fig. 2(c))). However, the difference in computation time between the cubic Hermite and the linear Lagrange elements is more pronounced in 3D than in 2D. When JFNK is used to solve the nonlinear system, the quadratic Lagrange elements have the shallowest slope on the log–log plot. In 2D (Fig. 2(b)), the cubic Hermite and the quadratic Lagrange elements exhibit a similar slope, while in 3D (Fig. 2(d)) the slope for the Hermite elements is the steepest.

Even though the quadratic Lagrange elements have the most DOFs, the Hermite elements are the most expensive. The reason for this can be identified from the values in Table 1. Though the quadratic Lagrange elements have more DOFs, they have fewer QPs and a smaller bandwidth, thus the overall computational expense for the Hermite elements is higher. The difference is smaller in 2D than in 3D because the difference between the number of QPs and the bandwidth is larger in 3D.

It is also interesting to directly compare the computational time using Newton’s method to that using JFNK. For a given number of elements, both Newton’s method and JFNK deliver the same solution (and thus the same error), however JFNK is takes more computation time. This is due to the computations needed to approximate the Jacobian. The true value of JFNK is that it delivers comparable results to those shown here even in cases when the full Jacobian is unavailable.
In order to identify which of our three approaches is the most efficient, we consider the relationship between computation time and error. To make this comparison, we combine the $L_2$-error from Fig. 1 with the computation time information from Fig. 2 on a single plot, shown in Fig. 3. This comparison is relevant since neither the computation time nor the overall accuracy is sufficient, in isolation, to demonstrate that a given discretization is more efficient than any of the others. For both 2D and 3D and using Newton’s method and JFNK, the linear Lagrange elements are clearly the least efficient approach. In 2D, the cubic Hermite elements are the most efficient, though the difference between the Hermites and the quadratic Lagrange elements is small. In 3D, the two higher-order elements are nearly identical, though their behavior is more similar using JFNK than with Newton’s method.

Fig. 1. Error comparison of the three solution approaches in 2D using Newton’s method. Discrete points and lines represent numerical results and corresponding linear fit, respectively. The linear Lagrange elements, quadratic Lagrange elements, and cubic Hermite elements deliver convergence rates of 2, 3, and 4, respectively, for the $L_2$-error (a) and rates of 1, 2 and 3 for the $H^1$-error (b). Note that the JFNK and 3D results are similar to those shown here.

Fig. 2. Computational time vs. number of elements along a side for 2D simulations using Newton’s method and JFNK ((a) and (b), respectively), and for 3D simulations also using Newton’s method and JFNK ((c) and (d), respectively). The third-order Hermite elements are the most expensive, while the first-order Lagrange elements are the least. The computation time reported is the total wall time for one time step, not including the startup time.
Having conducted a detailed investigation of the performance of the three approaches using a manufactured solution, we now demonstrate that consistent results are obtained when solving the CH equation to model microstructure evolution. The computational domain is a two-dimensional square of side length $l = 128.0$ with the double circle initial configuration shown in Fig. 4(a). The double circle and the surrounding area correspond to two phases with $c = 1$ and $c = -1$, respectively. The two circles coalesce over time to reduce surface area and eventually become a single circle. The three approaches used in the previous section are again applied to simulate this evolution process with an initial time step of $\Delta t = 1.0$. All simulations in this section were solved using Newton’s method. Since the microstructural evolution takes a long time and most transition happens at the beginning, the time step is adapted according to the scheme described in [14]. In this adaptive time step scheme, the integration error is estimated with a relative L2 norm computed by comparing the solution obtained after two time steps of size $\Delta t$ to the solution obtained with one time step of size $2\Delta t$. The time step adaptivity is based on a target

$$
\text{Computation time (s)} \quad \text{L2 error}
$$

<table>
<thead>
<tr>
<th>1st O. Lagrange</th>
<th>2nd O. Lagrange</th>
<th>3rd O. Hermite</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1.0 \times 10^{-6}$</td>
<td>$1.0 \times 10^{-4}$</td>
<td>$1.0 \times 10^{-2}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1st O. Lagrange</th>
<th>2nd O. Lagrange</th>
<th>3rd O. Hermite</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1.0 \times 10^{-6}$</td>
<td>$1.0 \times 10^{-4}$</td>
<td>$1.0 \times 10^{-2}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1st O. Lagrange</th>
<th>2nd O. Lagrange</th>
<th>3rd O. Hermite</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1.0 \times 10^{-6}$</td>
<td>$1.0 \times 10^{-4}$</td>
<td>$1.0 \times 10^{-2}$</td>
</tr>
</tbody>
</table>

**Fig. 3.** Computational efficiency (computation time vs. L2-error) comparison of the three approaches in 2D using Newton’s method and JFNK ((a) and (b), respectively), and in 3D using Newton’s method and JFNK ((c) and (d), respectively). The computation time reported is the total wall time for one time step, not including the startup time.

### 3.2. Two circle coalescence

Having conducted a detailed investigation of the performance of the three approaches using a manufactured solution, we now demonstrate that consistent results are obtained when solving the CH equation to model microstructure evolution. The computational domain is a two-dimensional square of side length $l = 128.0$ with the double circle initial configuration shown in Fig. 4(a). The double circle and the surrounding area correspond to two phases with $c = 1$ and $c = -1$, respectively. The two circles coalesce over time to reduce surface area and eventually become a single circle. The three approaches used in the previous section are again applied to simulate this evolution process with an initial time step of $\Delta t = 1.0$. All simulations in this section were solved using Newton’s method. Since the microstructural evolution takes a long time and most transition happens at the beginning, the time step is adapted according to the scheme described in [14]. In this adaptive time step scheme, the integration error is estimated with a relative L2 norm computed by comparing the solution obtained after two time steps of size $\Delta t$ to the solution obtained with one time step of size $2\Delta t$. The time step adaptivity is based on a target

$$
\text{Computation time (s)} \quad \text{L2 error}
$$

<table>
<thead>
<tr>
<th>1st O. Lagrange</th>
<th>2nd O. Lagrange</th>
<th>3rd O. Hermite</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1.0 \times 10^{-6}$</td>
<td>$1.0 \times 10^{-4}$</td>
<td>$1.0 \times 10^{-2}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1st O. Lagrange</th>
<th>2nd O. Lagrange</th>
<th>3rd O. Hermite</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1.0 \times 10^{-6}$</td>
<td>$1.0 \times 10^{-4}$</td>
<td>$1.0 \times 10^{-2}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1st O. Lagrange</th>
<th>2nd O. Lagrange</th>
<th>3rd O. Hermite</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1.0 \times 10^{-6}$</td>
<td>$1.0 \times 10^{-4}$</td>
<td>$1.0 \times 10^{-2}$</td>
</tr>
</tbody>
</table>

**Fig. 4.** Three time slices of the evolved double circle configuration computed using $128 \times 128$ elements: (a) $t = 0$, (b) $t = 2500$, and (c) $t = 10,000$. L. Zhang et al. / Journal of Computational Physics 236 (2013) 74–80
error tolerance etol = 0.001 and a maximum increase per time step of 10%. Note that the initial time step of \( \Delta t = 1.0 \) is sufficiently small to be below the target error for all three simulations.

The results from the previous section illustrated that for a given number of elements on a side \( N \), the higher order elements are more accurate but also significantly slower than the linear Lagrange elements. Therefore, we have chosen numbers of elements for each approach that give similar computation times. Again, the domain is uniformly discretized with \( N = 128 \) is used for the linear Lagrange elements, \( N = 64 \) is used for the quadratic Lagrange elements and \( N = 60 \) is used for the cubic Hermite elements.

\textbf{Figs. 4(b) and (c) show the simulation results with the linear Lagrange elements at } t = 2500 \text{ and } t = 10,000, \text{ respectively. To guarantee the three approaches deliver the same results for this transient problem, a more quantitative comparison between different types of elements is given in Fig. 5(a), which shows the evolution of the average concentration value along the length of the center line at } x = 64. \text{ It is found that the three types of elements are consistent, and the relative difference between solutions of different types of elements is less than 3\%. With the chosen mesh sizes for each approach, the computation times are similar (as shown in Fig. 5(b)).}

### 4. Conclusion and discussion

A detailed comparison of three solution techniques for the CH equation was conducted, using cubic Hermite elements and introducing an auxiliary variable in conjunction with linear and quadratic Lagrange elements. A series of quantitative MMS comparisons of different types of elements was conducted with the \textsc{Marmot/Moose} framework in the spatial domain using Newton’s method and JFNK. It was found that linear Lagrange elements have the lowest computational time, while cubic Hermite elements provide the most accuracy for a given \( N \). When both computational time and accuracy were considered, the cubic Hermite elements achieved the lowest error per unit of computational time in 2D, while in 3D the Hermite elements and the quadratic Lagrange elements were more closely matched in this metric. Finally, the microstructural evolution of a double circle was computed, and for this realistic application it was determined that the three types of elements predicted similar evolution in approximately the same computation time with \( N = 128, 64 \) and 60 elements in each direction for the linear Lagrange, the quadratic Lagrange and the cubic Hermite elements, respectively.

### Acknowledgments

The authors thank C. Permann (INL), J. Miller (INL), and R.H. Stogner (Univ. of Texas, Austin) for their invaluable assistance in the preparation of this manuscript. This work has been funded by the United States Nuclear Energy Advanced Modeling and Simulation program for Fundamental Methods and Modeling. The submitted manuscript has been authored by a contractor of the U.S. Government under Contract DE-AC07-05ID14517. Accordingly, the U.S. Government retains a non-exclusive, royalty free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.