An Unconditionally Stable One-Step Scheme for Gradient Systems

David J. Eyre*

June 9, 1998

Abstract

A semi-implicit time discretization of gradient flows is given and analyzed. It is shown that the scheme is unconditionally gradient stable and that the discrete equations are uniquely solvable for all time steps. The key feature of the method is a separation of the contractive and expansive terms of the equation across the time step. The Cahn-Hilliard equation is used as an example.

Keywords: Numerical stability, gradient flows, initial value problems

AMS subject classification: 65L05, 65L20

Running title: An unconditionally gradient stable scheme

*Department of Mathematics, University of Utah, Salt Lake City, UT 84112, eyre@math.utah.edu

1 Introduction.

This paper will concentrate on a numerical method for solving the initial value problem defined by the system of real ordinary differential equations and auxillary conditions

$$\frac{du}{dt} = -\nabla F(u), \quad u(0) = u_0. \tag{1}$$

It is assumed that $u(t) \in C^1(\mathbb{R}^+, \mathbb{R}^p)$, $F(u) \in C^2(\mathbb{R}^p, \mathbb{R})$, $\nabla F(u)$ is the gradient of F, and

$$\begin{cases} F(u) \geq 0 & \forall u \in \mathbb{R}^p \\ F(u) \to \infty & \text{as } \|u\| \to \infty \\ \langle J(\nabla F)(u)\, u,\, u\rangle \geq \lambda & \forall u \in \mathbb{R}^p \end{cases} \tag{2}$$

where $J(\nabla F)(u)$ is the Jacobian of $\nabla F(u)$ and $\lambda \in \mathbb{R}$, and where $\langle \cdot, \cdot \rangle$ denotes the inner product on $\mathbb{R}^p$ with the corresponding norm $\|u\|^2 = \langle u, u \rangle$. Such equations are called gradient systems, and their solutions are called gradient flows.

The analysis and results of this paper are given for ordinary differential equations, but the most interesting applications of the results are when the proposed method is used to solve discretized partial differential equations. Important examples of pdes that are gradient systems are the Cahn-Allen or bistable reaction diffusion equations [1], the Cahn-Hilliard equations [3], and the Ginzburg-Landau equations [4]. A discretized Cahn-Hilliard equation is used as an example in this paper.

All gradient systems have the property

$$\frac{dF(u)}{dt} = \langle \nabla F(u), u_t \rangle = -\|\nabla F(u)\|^2, \tag{3}$$

therefore, $F(u(t)) \leq F(u(0))$ for all $t \geq 0$. In the examples given above, $F(u)$ can be interpreted as an energy.

It is interesting to note that if $F(u)$ is convex, i.e. $\lambda \geq 0$, then only a single equilibrium exists for (1) and the flow is contractive. Unconditionally stable and uniquely solvable numerical schemes exist for these equations (see [9]). Alternatively, if $F(u)$ is not convex, then $\lambda < 0$, and multiple equilibria of (1) may exist and these gradient flows can expand in $u(t)$.

Notice that the third assumption on $F(u)$ in (2) is not restrictive because it bounds the rate that $u(t)$ can expand. Therefore, such a bound must necessarily exist for all gradient flows by (3).

This paper will concentrate on the case where $F(u)$ is not convex, i.e. $\lambda < 0$. In the numerical method, $F(u)$ will be split into the difference of two functions

$$F(u) = F_c(u) - F_e(u) \tag{4}$$

where

$$F_\circ \in C^2(\mathbb{R}^p, \mathbb{R}) \text{ and } F_\circ(u) \text{ is strictly convex for all } u \in \mathbb{R}^p. \tag{5}$$

Given a suitable splitting, notice that the flow of the equation $\dot{u} = -\nabla F_c(u)$ is contractive, and the flow of the equation $\dot{u} = \nabla F_e(u)$ is expansive.

In the following numerical approximation of $u(t)$, the splitting of $F(u)$ is exploited. Let U^n approximate $u(nk)$ where k is the time step, and require U^{n+1} to satisfy

$$U^{n+1} - U^n = k\left[\nabla F_e(U^n) - \nabla F_c(U^{n+1})\right], \tag{6}$$

where $U_0 = u_0$.

It has been argued [9] that it is unlikely that unconditionally stable numerical schemes for (1) and (2) exist because of the expansive and contractive nature of the flow. However, the goal of this paper is to show the following theorem for (6), and to apply (6) to the Cahn-Hilliard equation.

Theorem 1 *If $F_c(u)$ and $F_e(u)$ satisfy (4)-(5), and if $F_e(u)$ satisfies*

$$\langle [J(\nabla F_e)(u)]\, u,\, u\rangle \geq -\lambda \tag{7}$$

when $\lambda < 0$, then for any initial condition, the numerical scheme (6) is consistent, gradient stable for all $k > 0$ and possesses a unique solution for each time step. The local truncation error of each step is

$$\mathcal{T}^n = \frac{k^2}{2}\left[J(\nabla F_c)(\hat{u}) + J(\nabla F_e)(\hat{u})\right] \nabla F(u(\xi)) \tag{8}$$

for some $\xi \in (nk, (n+1)k)$ and for some $\hat{u}$ in the parallelopiped with opposite vertices at U^n and U^{n+1}.

The layout of the paper is as follows. In §2, gradient flows are discussed in more detail, and unconditional gradient stability is defined. In §3, some technical results and a proof of Theorem 1 are given. In §4 some splittings of the nonlinear terms are discussed. Finally, in §5 the Cahn-Hilliard equation is used for numerical tests of the method.

2 Gradient flows and gradient stability.

The behavior of gradient systems are characterized by the following, proved in [7].

Theorem 2 *For any solution $u(t)$ of (1) and (2) and any sequence $t_i \to \infty$ for which the $\omega-$limit point*

$$\bar{u} = \lim_{t\to\infty} u(t_i)$$

exists, it follows that $\bar{u} \in \mathcal{E}$, the set of zeros of ∇F. Furthermore if all members of $\mathcal{E}$ are isolated, then for each $u(0)$, there exists $\bar{u} \in \mathcal{E}$ such that

$$\lim_{t\to\infty} u(t_i) = \bar{u}.$$

For numerical approximations of gradient systems it is natural to seek a function of U that satisfies (2), replicates the energy dissipation property (3), and replicates the properties of Theorem 2. The following definition (modified from Stewart and Humphries in [9]) satisfies these requirements.

Definition 3 *A one-step numerical integration scheme is said to be unconditionally gradient stable if there exists a function $\mathcal{F}(\cdot) : \mathbb{R}^p \to \mathbb{R}$ such that, for all $k > 0$ and for all initial data:*

i. $\mathcal{F}(U) \geq 0$ *for all* $U \in \mathbb{R}^p$.

ii. $\mathcal{F}(U) \to \infty$ *as* $||U|| \to \infty$.

iii. $\mathcal{F}(U^{n+1}) \leq \mathcal{F}(U^n)$ *for all* $U^n \in \mathbb{R}^p$.

iv. If $\mathcal{F}(U^n) = \mathcal{F}(U_0)$ *for all* $n \geq 0$ *then* $U_0 \in \mathcal{E}$ *for (1) and (2).*

Notice that in the definition, $\mathcal{F}(U)$ and $F(U)$ are only required to share a few simple properites. A nice feature of (6) is that $\mathcal{F}(U) = F(U)$.

3 Proof of Theorem 1.

The following three lemmas are used in the proof of Theorem 1. They are related to the expansive properties of $F(u)$ and $F_e(u)$.

Lemma 4 *(Humphries [8]) Suppose $F(u)$ satisfies (2), then*

$$F(u) - F(v) \leq \langle \nabla F(u), u - v \rangle + |\lambda| \, \|u - v\|^2 \tag{9}$$

for any $u, v \in \mathbb{R}^p$.

Proof: With an exact Taylor Expansion of F about u up to second order, the result follows from the lower bound on the the negative eigenvalues of $J(\nabla F)$ given in (2). □

Lemma 5 *Suppose $F_e(u)$ satisfies (4)-(5), then there exists an interval, $\mathcal{I} = (0, \hat{\lambda})$, so that if $c \in \mathcal{I}$ then*

$$\langle \nabla F_e(u) - \nabla F_e(v), u - v \rangle \geq c \, \|u - v\|^2 \tag{10}$$

for any $u, v \in \mathbb{R}^p$.

Proof: By the Mean Value Theorem, it is sufficient to bound the eigenvalues of $J(\nabla F_e)$. Since $F_e(u)$ is convex, then $J(\nabla F_e)$ is positive definite and those eigenvalues are positive. □

Lemma 6 *Suppose $F_e(u)$ satisfies (4)-(5) and (7), and let $\hat{\lambda} = \sup \mathcal{I}$ for all u and v, then $\hat{\lambda} \geq |\lambda|$.*

Proof: By (7) and (4), all of the eigenvalues of $J(\nabla F_e)$ dominate the largest eigenvalue of $-J(\nabla F)$. □

Proof of Theorem 1: The unconditional gradient stability of (6) will be established first. Taking $\mathcal{F}(U) = F(U)$ implies that properties (i), (ii), and (iv) follow for the numerical approximation by the assumptions in (2) and by Theorem 2.

To show (iii), begin by applying Lemma 4 with $u = U^{n+1}$ and $v = U^n$ to the difference

$$\begin{aligned} F(U^{n+1}) - F(U^n) &\leq \langle \nabla F(U^{n+1}), U^{n+1} - U^n \rangle + |\lambda| \, \|U^{n+1} - U^n\|^2 \\ &= \langle \nabla F_c(U^{n+1}) - \nabla F_e(U^{n+1}), U^{n+1} - U^n \rangle + |\lambda| \, \|U^{n+1} - U^n\|^2 \end{aligned}$$

$$\begin{aligned}
& -\langle \tfrac{1}{k}\left[U^{n+1}-U^n\right] + \nabla F_c(U^{n+1}) - \nabla F_e(U^n), U^{n+1}-U^n\rangle \\
= \;& -\langle \nabla F_e(U^{n+1}) - \nabla F_e(U^n), U^{n+1}-U^n\rangle + (|\lambda| - \tfrac{1}{k})\,\|U^{n+1}-U^n\|^2 \\
\leq \;& -\hat{\lambda}\,\|U^{n+1}-U^n\|^2 + (|\lambda| - \tfrac{1}{k})\,\|U^{n+1}-U^n\|^2 \\
= \;& -(\hat{\lambda} - |\lambda| + \tfrac{1}{k})\|U^{n+1}-U^n\|^2 \\
\leq \;& 0
\end{aligned}$$

which follows by using (4) and (6), then gathering terms and applying Lemmas 5 and 6. This establishes (iii) for all k. Hence the method is unconditionally gradient stable.

The unique solvability of (6) at each step is now shown. The nonlinear equations

$$U^{n+1} + k\nabla F_c(U^{n+1}) = R^n \tag{11}$$

must be solved at each step for a given R^n. Since F_c is convex, then

$$\mathcal{G}(U^{n+1}) = \frac{1}{2}\|U^{n+1}\|^2 + kF_c(U^{n+1}) - \langle U^{n+1}, R^n\rangle$$

has a unique minimum for all k, and since $\nabla\mathcal{G} = 0$ is satisfied by (11), (6) has a unique solution for all $k \geq 0$.

The consistency and local truncation error of (6) can be established by standard local techniques using Taylor Series for U^{n+1} and U^n about $(n+\frac{1}{2})k$, and for $\nabla F_c(U^{n+1})$ and $\nabla F_e(U^n)$ about $U^{n+\frac{1}{2}}$. □

Remark: The local truncation error reduces to the local truncation error for Euler's method if $F(u) = F_e(u)$, and it reduces to the local truncation error for Backwards Euler's method if $F(u) = F_c(u)$. If both $F_c(u)$ and $F_e(u)$ are nonzero, the local truncation error of (6) is larger than the local truncation errors of Euler methods.

Remark: Similar results also hold for weighted gradient flows,

$$\frac{du}{dt} = -A\nabla F(u)$$

where A is symmetric, positive semi-definite and left invertible, and with the numerical scheme

$$U^{n+1} - U^n = kA\left[\nabla F_e(U^n) - \nabla F_c(U^{n+1})\right].$$

This scheme is consistent, unconditionally gradient stable and U^{n+1} exists uniquely. The proofs of these statements follow from straightforward modifications of the proof of Theorem 1.

Remark: Since $\mathcal{G}(U^{n+1})$ is convex, if Newton's method is used to find U^{n+1}, it will converge to U^{n+1} from any initial guess.

4 The splitting of $F(u)$.

The obvious weakness of this method is that the splitting must be calculated. This requires a substantial amount of knowledge about $F(u)$, and finding a splitting of $F(u)$ is a difficult task because the decomposition $F(u) = F_e(u) - F_c(u)$, which always exists, is not unique [2]. Beyond the fact that the splitting is not unique, the criteria for choosing a splitting may also depend on the equations that are to be solved. Three reasonable criteria for choosing a splitting are, first, the splitting should be easy to calculate, second, the splitting should be chosen so that the local truncation error is as small a possible, and third, the splitting should be chosen so the update equations are easy to solve.

The next three subsections give examples that satisfy these last three conditions. These are not the only splittings that can be found, but the ideas could be applied to any function.

4.1 A simple splitting of $F(u)$.

An example of a splitting that statisfies (5) and (7) is to set

$$F_e(u) = |\lambda|\langle u, u\rangle. \tag{12}$$

In this splitting, the expansive terms of $F(u)$ are approximated by a simple quadratic that is everywhere less than $F(u)$.

To find λ, it would be necessary to find the global minimum of $\langle J(\nabla F)(u)\, u, u\rangle$ over $u \in \mathbb{R}^p$. Alternatively, if the global minimum is difficult to find, but if it can be easily bounded from from below by some number $-C \leq \lambda$, then $F_e(u) = C\langle u, u\rangle$ also satisfies the assumptions of Theorem 1 but will give a larger local truncation error.

4.2 A splitting of $F(u)$ with small truncation error.

A splitting where the terms of $F(u)$ are divided into contractive and expansive terms and $F_c(u)$ and $F_e(u)$ are constructed as a sum of the contractive and expansive terms, respectively, will generically have the smallest possible local truncation error. This follows because no terms are added and subtracted to $F(u)$. This splitting is employed in the following example.

Example: Consider the equation $du/dt = u - u^3$ where $p = 1$ and $F(u) = (u^2 - 1)^2/4$. A splitting of $F(u)$ that satisfies (4)-(5) is

$$F_c(u) = \frac{u^4}{4} + 1 \quad \text{and} \quad F_e(u) = \frac{u^2}{2}.$$

Then (6) becomes

$$U^{n+1} + k(U^{n+1})^3 = (1+k)U^n.$$

Notice that U^{n+1} is a root of a cubic polynomial, and on the real numbers, that polynomial is monotone increasing for all $k > 0$. Therefore, U^{n+1} is uniquely defined.

Caution should be used with this separation of expansive and contractive terms since it may not satisfy the spectral condition (7).

This separation of expansive and contractive terms requires some individual terms to be separated across the time step. For example, suppose that $p = 1$ and that $F(u)$ contains the term u^3, then $-\nabla F(u)$ contains the term $-u^2$. This term is contractive if $u(t)$ is positive and expansive if $u(t)$ is negative. A reasonable splitting in this case is to employ the Heaviside function $H(u)$ and to calculate $H(U^{n+1})(U^{n+1})^2$ implicitly and $H(-U^n)(U^n)^2$ explicitly.

4.3 A linear splitting of $F(u)$.

A splitting that results in update equations that are easy to solve exploits the following corollary of Theorem 1.

Corollary 7 *Let* $\mathcal{R}(v) = \{u \in \mathbb{R}^p | F(u) \leq F(v)\}$, *then* $U^n \in \mathcal{R}(U_0)$ *for all* $n \geq 0$.

In many cases, $\mathcal{R}(u_0)$ is easy to find. This observation can be used to find a splitting of $F(u)$ so that the right hand side of (6) is a *linear* equation for U^{n+1}, i.e.

$$(I + kA_c)U^{n+1} = U^n + k\nabla F_e(U^n)$$

where A_c is a symmetric constant coefficient matrix.

To find a suitable matrix A_c, $F_e(u) = \frac{1}{2}\langle A_c\, u, u\rangle - F(u)$ must be convex and statisfy (7). Therefore, it is sufficient to choose

$$A_c = J(\nabla F)(u^*) + \sigma I \tag{13}$$

where u^* is chosen so that

$$\langle u^*, [J(\nabla F)(u^*)]u^*\rangle \geq \langle u, [J(\nabla F)(u)]u\rangle \quad \forall u \in \mathcal{R}(u_0). \tag{14}$$

and σ is chosen to satisfy (7) on $\mathcal{R}(u_0)$. In this case,

$$\langle [A_c - J(\nabla F)(u)]\; u, u\rangle \geq 0, \qquad \forall u \in \mathcal{R}(u_0).$$

thus, $F_c(u)$ and $F_e(u)$ are convex by construction. The associated cost of this scheme is a larger local truncation error.

Example: Consider again $du/dt = u - u^3$ with $|u(0)| < 1$. From the graph of $F(u)$, it is easy to see that $\mathcal{R}(u_0) \subset \mathcal{R}(0)$. Since $F(0) = F(\pm\sqrt{2}) = \frac{1}{4}$, the interval $(-\sqrt{2}, \sqrt{2})$ can be used to calculate $A_c = 5$ from $F''(u) = 1 - 3u^2$. Then a splitting of $F(u)$ that satisfies (4)-(5) and (13) is

$$F_c(u) = \frac{5u^2}{2} + 1 \text{ and } F_e(u) = \frac{6u^2}{2} - \frac{u^4}{4},$$

and (6) becomes

$$(1 + 5k)U^{n+1} = (1 + 6k)U^n - k(U^n)^3.$$

In fact, it is easy to see $u \in [-1, 1]$, and using $A_c = 2$ the scheme is also gradient-stable on that interval. This latter choice will give a smaller local truncation error.

Remark: It may be desirable in certain cases to take A_c to be a diagonal matrix or a scalar. Such a choice would increase the local truncation error, but it may significantly decrease the effort of solving the linear equations. This may be particularly true for time stepping systems of partial differential equations because the coupling of the equations may be significantly decreased.

5 An example: The Cahn-Hilliard Equation

In this section, the Cahn-Hilliard (CH) equation will be used to demonstrate the method. The Cahn-Hilliard equation is a gradient flow of

$$F(u) = \int_0^\ell \left[\frac{\varepsilon^2}{2} u_x^2 + W(u)\right] dx \tag{15}$$

where $\nabla F(u)$ is interpreted as a constrained gradient with respect to changes in u in the Hilbert space $H^{-1}(0,\ell)$. The scaling parameter satisfies $0 < \varepsilon \ll 1$, and $W(u)$ is a double well potential bounded from below, typically $W(u) = (u^2-1)^2/4$. For a review on Cahn-Hilliard theory, see [6].

For the given nonlinearity, the Cahn-Hilliard equation in one dimension is

$$\frac{\partial u}{\partial t} = -\varepsilon^2 u_{xxxx} + (u^3)_{xx} - u_{xx}.$$

The contractive terms of the equation are $-\varepsilon^2 u_{xxxx}$ and $(u^3)_{xx}$, and the expansive term is backwards heat operator $-u_{xx}$.

The appropriate boundary conditions are

$$u_x(0,t) = u_x(\ell,t) = u_{xxx}(0,t) = u_{xxx}(\ell,t) = 0,$$

and a natural physical problem suggests that $u(x,0)$ is random on a small interval about a specified average.

The dynamics of the equation take place on numerous time scales, and some of the more interesting dynamics of the equation take place on short time scales $t = O(\varepsilon)$ and on exceedingly long time scales $t = O(e^{1/\varepsilon})$. During the short time scale, the nearly uniform solution develops a fine structure with numerous internal layers and sharp gradients. On the longer time scales, the number of internal layers decreases as the solutions coarsen.

The numerous time scales make the discrete equations stiff. Additionally, the spatial step size must be small because within the internal layers, $u_x = O(1/\varepsilon)$. Thus, to resolve the layers in numerical calculations, the spatial discretization must be $h = O(\varepsilon)$.

Since the spatial discretization is fine, both the biharmonic operator and the nonlinearity are stiff, so explicit and semi-implicit schemes have restrictive time steps. Alternatively, the time step for implicit schemes is restricted by solvability troubles due the $-u_{xx}$ term.

To present the results, the partial differential equation will be approximated by a system of ordinary differential equations obtained by discretizing the Cahn-Hilliard equation in space. To this end, let $u_j = u_j(t)$ approximate $u(jh, t)$ where $h = \ell/M$ is the spatial discretization parameter. Furthermore, let the laplacian operator be approximated by the standard 3 point stencil, i.e.

$$u_{xx}(jh, t) \to \frac{1}{h^2}\left(u_{j-1}(t) - 2u_j(t) + u_{j+1}(t)\right)$$

and denote the discrete laplacian by D_h. In this case, the Cahn-Hilliard equation is reduced to $M+1$ ordinary differential equations of the form

$$\frac{du_j}{dt} = -D_h\left[\varepsilon^2 D_h\, u_j + (u_j^3) - u_j\right]$$

for $j = 0, 1, ..., M$. The boundary conditions can be used to find that $u_{-1}(t) = u_1(t)$, $u_{-2}(t) = u_2(t)$, $u_{M+1}(t) = u_{M-1}(t)$ and $u_{M+2}(t) = u_{M-2}(t)$. This system of equations can be thought of as a weighted gradient flow of a discrete analog of (15).

Five methods of time stepping the Cahn-Hilliard equation will now be considered. Let U_j^n approximate $u(jh, nk)$, then the five methods to find U_j^{n+1} are;

1. Explicit Euler's scheme

$$U_j^{n+1} - U_j^n = -\varepsilon^2 k D_h^2\, U_j^n + k D_h\,(U_j^n)^3 - k D_h\, U_j^n$$

2. Semi-implicit Euler's scheme

$$U_j^{n+1} - U_j^n = -\varepsilon^2 k D_h^2\, U_j^{n+1} + k D_h\,(U_j^n)^3 - k D_h\, U_j^n$$

3. Implicit Euler's scheme

$$U_j^{n+1} - U_j^n = -\varepsilon^2 k D_h^2 U_j^{n+1} + k D_h (U_j^{n+1})^3 - k D_h U_j^{n+1}$$

4. Non-linearly stabilized splitting scheme

$$U_j^{n+1} - U_j^n = k \left(-\varepsilon^2 D_h^2 U_j^{n+1} + D_h (U_j^{n+1})^3 \right) - k D_h U_j^n$$

5. Linearly stabilized splitting scheme

$$U_j^{n+1} - U_j^n = k \left(-\varepsilon^2 D_h^2 U_j^{n+1} + 2 D_h U_j^{n+1} \right) + k \left(D_h (U_j^n)^3 - 3 D_h U_j^n \right)$$

In all cases, $\ell = 1$, $\varepsilon = 0.01$ and $h = \varepsilon/2 = 0.005$. A banded LU decomposition of the penta-diagonal matrix was calculated when needed. The calculations were performed in Matlab which offers easy access to the number of floating point operations (flops).

The explicit Euler's scheme is not gradient stable, and in numerical tests, it was found that the stable time steps are restricted by $k \le 8 \times 10^{-7}$. Using the maximum time step the scheme requires 4.42×10^9 flops for each unit interval of time.

The semi-implicit Euler's scheme is much better, but still not gradient stable. This scheme is also reasonably simple, and in numerical tests, it was found that $k \le 2 \times 10^{-4}$ and the scheme requires 3.87×10^7 flops for each unit interval of time. So this scheme requires 74 times less computational effort than Euler's scheme.

The implicit Euler's scheme is known to be gradient stable for the Cahn-Hilliard equation with $k \le h^2/4 = 6.25 \times 10^{-6}$, [5] [6] [9]. The allowed time step in this case is only 8 times larger than Euler's scheme. Yet there is another restriction. To have a unique solution, the linearized equations used by Newton's method must be positive definite. To ensure this condition, Gershgoren's Theorem can be applied to the linearized system. This restricts the time step to $k \le h^2/18 = 1.39 \times 10^{-6}$; less than twice the explicit Euler's step. Therefore, the expense of Newtons method to solve the resulting nonlinear equations will make the cost of this scheme exceed the cost of the explicit Euler's scheme. As a side note, the Crank-Nicholson scheme also suffers from this solvability restriction, and in this case $k \le h^2/9$.

The nonlinearly stablized scheme takes

$$F_c = \int_0^1 \left[\frac{\varepsilon^2}{2} u_x^2 + \frac{u^4}{4} + 1 \right] dx \text{ and } F_e = \frac{1}{2} \int_0^1 u^2 \, dx$$

for the splitting (4). Newton's method was employed to solve the nonlinear equations. The stopping criteria for Newton's method was chosen to be

$$\|r_m\| \le 10^{-7}$$

where r_m denotes the residue at iterate m. The solution at time level n was taken as the initial guess for Newton's method.

Adaptive stepping was used to bound an estimated local truncation error below 10^{-5}. The average time step on the interval $(0,1)$ was $h_{ave} = 3.3 \times 10^{-3}$ and the number of flops required was 5.83×10^6. The average number of Newton iterates required for convergence was 2.3.

Over this time interval and with the given initial conditions, the Cahn-Hilliard equation has fast dynamics. However, even in the presence of these fast dynamics, the nonlinearly stabilized method requires roughly 8 times less work that the semi-implicit method.

This observation suggests that the first three schemes were bound by stability restrictions, and not by local error considerations.

However, the strength of the stabilized method is more pronounced once the internal layers of the solution have fully developed. In this time frame, the solution changes very slowly. While the first three schemes will still be restricted to small time steps, the nonlinearly stabilized scheme can take much longer steps. An adaptive stepping scheme based on Richardson extrapolation suggested some step sizes exceeding 10^4. The following table summarizes the computational effort for large fixed time steps.

k	iterations/step	flops/step
1	2	28653
10	3	35737
100	4	49497
1000	6	63257
10000	9	90777

Notice that in the smallest step case, $k = 1$, this method requires roughly 1300 times less effort than the semi-implicit scheme.

To find a linearly stabilized scheme, it is known that the solution generally resides in the interval $(-1, 1)$, so the region

$$\mathcal{R} = \{u : |u| \leq 1\}$$

was chosen. The linear splitting for (4)-(5) uses

$$F_c = \int_0^1 \left[\frac{\varepsilon^2}{2}u_x^2 + u^2 + 1\right] dx \text{ and } F_e = \int_0^1 \left[\frac{u^4}{4} - \frac{3u^2}{2}\right] dx$$

to arrive at the specified scheme.

This method is linear, so it is two to three times faster than the nonlinear scheme because only one linear solve is required per step. Each step of this method requires nearly the same effort as one step of the semi-implicit scheme, but this method can take longer time steps for equal effort. The computational effort required to compute one step is identical for all step sizes because no iteration is performed. One step required 10885 flops, which over long time steps again represents a significant decrease in computational work.

Adaptive stepping was used to keep an estimated local truncation error below 10^{-5}. The average time step on the interval $(0, 1)$ was $h_{ave} = 2.7 \times 10^{-3}$ and the number of flops required was 1.29×10^6.

The flops reported in this paper do not directly apply to solving the Cahn-Hilliard equation in more than one spatial dimension because linear system solvers are more complicated, and maybe ill-conditioned. However, the stabilized time stepping schemes should clearly decrease the effort in time marching. In fact, they may make the use of iterative solvers more efficient because the matrices in the stabilized scheme are guaranteed to be positive definite and have eigenvalues that exceed one.

References

[1] S. Allen and J. W. Cahn, *A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening*, Acta Metallurgica, 27 (1979), pp. 1084-1095.

[2] H. Busemann, *Convex Surfaces*, Interscience Publishers, New York, 1958.

[3] J. W. Cahn, *On spinodal decomposition*, Acta Metallurgica, 9 (1961), pp. 795–801.

[4] J. Chapman, S. Howison, and J. Okendon, *Macroscopic models for superconductivity*, SIAM Rev., 34 (1992), pp. 529-560.

[5] Q. Du and R. Nicolaides, *Numerical analysis of a continuum model of phase transition*, SIAM J. Num. Anal., 28, (1991), pp. 1310-1322.

[6] C. M. Elliott, *The Cahn-Hilliard model for the kinetics of phase separation*, in Mathematical Models for Phase Change Problems, J. F. Rodrigues, ed., Birkhäuser Verlag, Basel, 1989, pp. 35–73.

[7] M. Hirsch and S. Smale, Differential Equation, Dynamical Systems and Linear Algebra, Academic Press, London, 1974.

[8] A. R. Humphries, Ph.D. Thesis, University of Bath, 1994.

[9] A. M. Stuart and A. R. Humphries, *Model problems in numerical stability theory for initial value problems*, SIAM Rev., 36 (1994), pp. 226-257.