

High order, semi-implicit, energy stable schemes for gradient flows

Alexander Zaitzeff^{a,*}, Selim Esedoğlu^a, Krishna Garikipati^b

^a*Department of Mathematics, University of Michigan, Ann Arbor, MI 48109, USA*

^b*Departments of Mechanical Engineering, and Mathematics, Michigan Institute for Computational Discovery & Engineering, University of Michigan, Ann Arbor, MI 48109, USA*

Abstract

We introduce a class of high order accurate, semi-implicit Runge-Kutta schemes in the general setting of evolution equations that arise as gradient flow for a cost function, possibly with respect to an inner product that depends on the solution, and we establish their energy stability. This class includes as a special case high order, unconditionally stable schemes obtained via convexity splitting. The new schemes are demonstrated on a variety of gradient flows, including partial differential equations that are gradient flow with respect to the Wasserstein (mass transport) distance.

1. Introduction

We are concerned with numerical schemes for evolution equations that arise as gradient flow (steepest descent) for an energy $E : H \rightarrow \mathbb{R}$, where H is a Hilbert space with inner product $\langle \cdot, \cdot \rangle$:

$$u' = -\nabla_H E(u). \quad (1.1)$$

Additionally, we will study gradient flows with a solution dependent inner product:

$$u' = -\mathcal{L}(u)\nabla_H E(u) \quad (1.2)$$

*Corresponding author

Email addresses: azaitzef@umich.edu (Alexander Zaitzeff), esedoglu@umich.edu (Selim Esedoğlu), krishna@umich.edu (Krishna Garikipati)

where $\mathcal{L}(u)$ is a positive definite operator that depends on u .

Equations (1.1) and (1.2) may represent (scalar or vectorial) ordinary or partial differential equations. One property of (1.1) and (1.2) is dissipation: $\frac{d}{dt}E(u) \leq 0$, to see this

$$\frac{d}{dt}E(u) = \langle \nabla_H E(u), u' \rangle = -\langle \nabla_H E(u), \mathcal{L}(u) \nabla_H E(u) \rangle \leq 0.$$

In [1], the authors focused on unconditionally stable numerical methods to solve (1.1). In this paper, our focus is on semi-implicit methods that come with a rigorously established energy stability property inherited from the above expression of dissipation. Specifically, let

$$E(u) = E_1(u) + E_2(u) \tag{1.3}$$

where in our numerical implementation we will handle E_1 implicitly and E_2 explicitly. We allow any choice for E_1 and E_2 as long as (1.3) is satisfied. Our numerical methods will guarantee that when the time step is less than a constant depending only on E_2 the following numerical dissipation property will hold:

$$E(u_{n+1}) \leq E(u_n) \tag{1.4}$$

where u_n denotes the approximation to the solution at the n -th time step.

A common, but typically only first order accurate, semi-implicit scheme for the abstract equation (1.1), with time step size $k > 0$, reads

$$\frac{u_{n+1} - u_n}{k} = -\nabla_H E_1(u_{n+1}) - \nabla_H E_2(u_n). \tag{1.5}$$

Let $L_2(u, u_n)$ be the linearization of E_2 around u_n so

$$L_2(u, u_n) = E_2(u_n) + \langle \nabla_H E_2(u_n), u - u_n \rangle.$$

Then (1.5) is the Euler-Lagrange equation for the optimization problem

$$u_{n+1} = \arg \min_u E_1(u) + L_2(u, u_n) + \frac{1}{2k} \|u - u_n\|^2 \tag{1.6}$$

where $\|\cdot\|^2 = \langle \cdot, \cdot \rangle$. For

$$\Lambda = \max\{0, \max_{u, \|v\|=1} D^2 E_2(u)(v, v)\}, \tag{1.7}$$

where by $D^2 E_2(u)(v, w)$ we mean $\frac{d^2}{d\epsilon_2 d\epsilon_1} E_2(u + \epsilon_1 v + \epsilon_2 w) \Big|_{\epsilon_1 = \epsilon_2 = 0}$, we have

$$E_2(u) \leq L_2(u, p) + \frac{\Lambda}{2} \|u - p\|^2 \quad (1.8)$$

for any u and p . It follows that when $k \leq \frac{1}{\Lambda}$,

$$\begin{aligned} E(u_{n+1}) &= E_1(u_{n+1}) + E_2(u_{n+1}) \leq E_1(u_{n+1}) + L_2(u_{n+1}, u_n) + \frac{1}{2k} \|u_{n+1} - u_n\|^2 \\ &\leq E_1(u_n) + L_2(u_n, u_n) + \frac{1}{2k} \|u_n - u_n\|^2 = E_1(u_n) + E_2(u_n) = E(u_n), \end{aligned}$$

so that scheme (1.5) is stable under this condition on the time step k , provided that optimization problem (1.6) can be solved. In practice, this discussion requires finding an upper bound on Λ in (1.7). Sometimes this is trivial, for instance when E_2 is quadratic. In many other cases, energy dissipation (1.4) can ensure solutions generated by scheme (1.5) remain small in an appropriate sense, giving an easy upper bound on Λ , which in turn guarantees continued energy dissipation; this is true, for instance, in the case of schemes for the 1D Allen-Cahn and Cahn-Hilliard equations discussed in section 5 as numerical examples. In any case, we assume there is already a scheme of the form (1.5) that is of practical interest, save for its low order of accuracy – this is our starting point. One of our main goals in this paper is to jack up the order of accuracy of such an existing semi-implicit scheme in a universal (problem independent) and painless fashion, by essentially calling the same code used to implement the first order scheme (1.5) itself multiple times per time step, and at the expense of replacing the stability parameter (1.7) by one that is not substantially worse or harder to bound. In particular, unconditionally stable schemes of the form (1.5) remain so.

Our new class of methods has no assumption (e.g. convexity, concavity) on the components E_1 and E_2 of the energy that are treated implicitly and explicitly, respectively. We note that depending on the problem, $E_1(u)$ may need to be convex to ensure the unique solvability in (1.5). But this is not always necessary, for example see [2] where $E_1(u)$ is concave and $E_2(u) = 0$. Our methods do not impose constraints on $E_1(u)$ and $E_2(u)$ past what is needed to

solve (1.5) and have at least up to third order accuracy. Our stability results are conditional, but revert to unconditional stability when E_1 and E_2 have appropriate convexity properties, and contain as a special case previous unconditional stability results for high order, convexity splitting type schemes. In particular, the previous papers [3, 4] on higher order ARK IMEX energy stable schemes study, in the spirit of convexity splitting [5], formulations that break up the energy into convex and concave parts, and treat the convex part implicitly and the concave part explicitly. We know of no other work on problem independent ARK IMEX methods that ensures energy stability in the very strong sense of (1.4).

It should be mentioned that numerical schemes for gradient flows and their stability, particularly in the context of phase-field models such as Allen-Cahn and Cahn-Hilliard equations that are included as examples in the present paper, have a long history. Since in practice one is interested in simulating the dynamics of these equations for very long times, numerical methods with robust stability properties, such as implicit and semi-implicit schemes, together with ideas such as convexity splitting, have attracted much attention, see e.g. [4, 6, 7, 8]. Many earlier works develop and study schemes in the context of specific equations and models, such as the phase-field crystal equations in addition to various versions of Allen-Cahn and Cahn-Hilliard, sometimes with complete proofs of convergence e.g. [9, 10, 11, 12, 13]. They include second order accurate in time methods (e.g. [14, 15, 16, 17, 18, 19, 20]), as well as third order in time methods, e.g. [21, 22, 23]. While the high order, energy stable (in the very strong sense of (1.4)) multi-stage schemes we develop in this paper are problem independent, going further to establish convergence based on these important consistency and stability properties may of course require using additional information about the specific equation being approximated.

Another goal of the present paper is extending these high order, stable, implicit and semi-implicit methods for general gradient flows to solve (1.2), the case when the inner product is solution dependent. There are certainly existing stable methods for (1.2) on a case by case basis, for example for the

Cahn-Hilliard equation with degenerate mobility [24, 25] and the porous medium equation [26, 27, 28]. To our knowledge, this is the first time high order multi-stage schemes for general gradient flows with solution dependent inner products have been developed that are guaranteed to dissipate the original energy from one time step to the next, as in (1.4).

The rest of the paper is organized as follows:

- Section 2 presents the conditions for energy stability and constructs schemes that satisfy them.
- In section 3, we state the consistency equations for the ARK IMEX schemes for solving gradient flows (1.1) and give 2nd and 3rd order examples.
- Section 4 gives 2nd and 3rd order methods for solving gradient flows with solution dependent inner product (1.2) and provides consistency calculations.
- In section 5, we present numerical convergence studies of several well-known partial differential equations that are gradient flows, including with respect to Wasserstein metrics.

The code for section 5 is publicly available, and can be found at <https://github.com/AZaitzeff/SIgradflow>.

2. Stability of Our New Schemes

In this section, we formulate a wide class of numerical schemes that are energy stable by construction. The first of these schemes are Implicit-Explicit Additive Runge-Kutta (ARK IMEX) schemes, but we will write them in variational form in order to prove energy stability more easily. The variational formulation of our M -stage ARK IMEX scheme is:

1. Set $U_0 = u_n$.

2. For $m = 1, \dots, M$:

$$U_m = \arg \min_u \left(E_1(u) + \sum_{i=0}^{m-1} \theta_{m,i} L_2(u, U_i) + \sum_{i=0}^{m-1} \frac{\gamma_{m,i}}{2k} \|u - U_i\|^2 \right). \quad (2.1)$$

where

$$L_2(u, p) = E_2(p) + \langle \nabla_H E_2(p), u - p \rangle. \quad (2.2)$$

3. Set $u_{n+1} = U_M$.

We assume that (2.1) has a unique solution at every step; just like for the original first order scheme (1.5) & (1.6) that is our starting point, this may in general entail a restriction on the time step size k . We reiterate that the energy is defined over an *arbitrary* Hilbert space, H . The norm in (2.1) is with respect to the inner product of H .

The schemes for approximating a gradient flow with respect to a *solution dependent inner product* will be a series of embedded ARK IMEX methods. The inner product will be fixed for each ARK IMEX step, allowing the stability results of this section to apply. Now we establish quite broad conditions on the coefficients $\gamma_{m,i}$ and $\theta_{m,i}$ that ensure conditional energy dissipation (1.4). Before we state and prove the conditions in generality, consider the following two-stage special case of scheme (2.1):

$$U_1 = \arg \min_u \left(E_1(u) + L_2(u, u_n) + \frac{\gamma_{1,0}}{2k} \|u - u_n\|^2 \right), \quad (2.3)$$

$$u_{n+1} = \arg \min_u \left(E_1(u) + \theta_{2,1} L_2(u, U_1) + \theta_{2,0} L_2(u, u_n) + \frac{\gamma_{2,0}}{2k} \|u - u_n\|^2 + \frac{\gamma_{2,1}}{2k} \|u - U_1\|^2 \right) \quad (2.4)$$

Let $\Lambda = \max\{0, \max_{u, \|v\|=1} D^2 E_2(u)(v, v)\}$. Note that this implies

$$E_2(u) \leq L_2(u, p) + \frac{\Lambda}{2} \|u - p\|^2 \quad (2.5)$$

for any u and p . Also note that $L_2(u, u) = E_2(u)$. Impose the conditions

$$\begin{aligned} \gamma_{1,0} - k\Lambda - \frac{(\gamma_{2,0} - k\Lambda\theta_{2,0})^2}{(\gamma_{2,0} + \gamma_{2,1} - k\Lambda\theta_{2,0} - k\Lambda\theta_{2,1})} &\geq 0, \\ \gamma_{2,1} + \gamma_{2,0} - k\Lambda\theta_{2,0} - k\Lambda\theta_{2,1} &> 0, \\ \theta_{2,1} + \theta_{2,0} &= 1 \text{ and} \\ \theta_{2,1}, \theta_{2,0} &\geq 0 \end{aligned} \quad (2.6)$$

on the parameters. Set $\mu = \frac{\gamma_{2,0} - k\Lambda\theta_{2,0}}{\gamma_{2,0} + \gamma_{2,1} - k\Lambda\theta_{2,0} - k\Lambda\theta_{2,1}}$. First note that (2.4) is equivalent to

$$\begin{aligned} u_{n+1} = \arg \min_u & E_1(u) + \theta_{2,1}L_2(u, U_1) + \theta_{2,0}L_2(u, u_n) + \theta_{2,1}\frac{\Lambda}{2}\|u - U_1\|^2 + \\ & \theta_{2,0}\frac{\Lambda}{2}\|u - u_n\|^2 + \frac{\gamma_{2,0} + \gamma_{2,1} - k\Lambda\theta_{2,0} - k\Lambda\theta_{2,1}}{2k}\|u - (\mu u_n + (1 - \mu)U_1)\|^2. \end{aligned} \quad (2.7)$$

This can be seen by expanding the norm squared and comparing the quadratic and linear terms in u . With these tools in hand we can prove energy dissipation:

$$\begin{aligned} & E(u_{n+1}) \\ &= E_1(u_{n+1}) + E_2(u_{n+1}) \\ &\leq E_1(u_{n+1}) + \theta_{2,1}[L_2(u_{n+1}, U_1) + \frac{\Lambda}{2}\|u_{n+1} - U_1\|^2] \\ &\quad + \theta_{2,0}[L_2(u_{n+1}, u_n) + \frac{\Lambda}{2}\|u_{n+1} - u_n\|^2] \quad (\text{by (2.5)}) \\ &\leq E_1(u_{n+1}) + \theta_{2,1}[L_2(u_{n+1}, U_1) + \frac{\Lambda}{2}\|u_{n+1} - U_1\|^2] \\ &\quad + \theta_{2,0}[L_2(u_{n+1}, u_n) + \frac{\Lambda}{2}\|u_{n+1} - u_n\|^2] \\ &\quad + \frac{\gamma_{2,0} + \gamma_{2,1} - k\Lambda\theta_{2,0} - k\Lambda\theta_{2,1}}{2k}\|u_{n+1} - (\mu u_n + (1 - \mu)U_1)\|^2. \quad (\text{by (2.6)}) \\ &\leq E_1(U_1) + \theta_{2,1}E_2(U_1) + \theta_{2,0}[L_2(U_1, u_n) + \frac{\Lambda}{2}\|U_1 - u_n\|^2] \\ &\quad + \frac{\gamma_{2,0} + \gamma_{2,1} - k\Lambda\theta_{2,0} - k\Lambda\theta_{2,1}}{2k}\|U_1 - (\mu u_n + (1 - \mu)U_1)\|^2 \quad (\text{by (2.7)}) \\ &\leq E_1(U_1) + \theta_{2,1}[L_2(U_1, u_n) + \frac{\Lambda}{2}\|U_1 - u_n\|^2] \\ &\quad + \theta_{2,0}[L_2(U_1, u_n) + \frac{\Lambda}{2}\|U_1 - u_n\|^2] \\ &\quad + \frac{(\gamma_{2,0} - k\Lambda\theta_{2,0})^2}{(\gamma_{2,0} + \gamma_{2,1} - k\Lambda\theta_{2,0} - k\Lambda\theta_{2,1})2k}\|U_1 - u_n\|^2 \quad (\text{by (2.5)}) \\ &\leq E_1(U_1) + [L_2(U_1, u_n) + \frac{\Lambda}{2}\|U_1 - u_n\|^2] + \frac{\gamma_{1,0} - k\Lambda}{2k}\|U_1 - u_n\|^2 \quad (\text{by (2.6)}) \\ &\leq E(u_n). \quad (\text{by (2.3)}) \end{aligned}$$

The first two conditions of (2.6) require k to be below a certain threshold. Hence the dissipation of (2.3) & (2.4) is conditional, unless E_2 happens to be concave in which case these two conditions are satisfied for all $k > 0$.

We will now extend this discussion to general, M -stage case of scheme (2.1):

Theorem 2.1. Fix a time step k . Define $\Lambda = \max\{0, \max_{u, \|v\|=1} D^2 E_2(u)(v, v)\}$ and the following auxiliary quantities in terms of the coefficients $\gamma_{m,i}$ and $\theta_{m,i}$ of scheme (2.1):

$$\tilde{\gamma}_{m,i} = \gamma_{m,i} - k\Lambda\theta_{m,i} - \sum_{j=m+1}^M \tilde{\gamma}_{j,i} \frac{\tilde{S}_{j,m}}{\tilde{S}_{j,j}}, \quad (2.8)$$

$$\tilde{S}_{j,m} = \sum_{i=0}^{m-1} \tilde{\gamma}_{j,i} \quad (2.9)$$

If $\tilde{S}_{m,m} > 0$ for $m = 1, \dots, M$, $\theta_{m-1,i} \geq \theta_{m,i} \geq 0$ and $\sum_{i=0}^{m-1} \theta_{m,i} = 1$, then scheme (2.1) satisfies the energy stability condition (1.4): For every $n = 0, 1, 2, \dots$ we have $E(u_{n+1}) \leq E(u_n)$.

As we will see in section 3, the conditions on the parameters $\gamma_{i,j}$ and $\theta_{m,i}$ of scheme (2.1) imposed in theorem 2.1 are loose enough to enable meeting consistency conditions to high order. We will establish theorem 2.1 with the help of a couple of lemmas:

Lemma 2.2. Let the auxiliary quantities $\tilde{S}_{j,m}$, and $\tilde{\gamma}_{m,i}$ be defined as in theorem 2.1. We have

$$\begin{aligned} & \arg \min_u E_1(u) + \sum_{i=0}^{m-1} \theta_{m,i} L_2(u, U_i) + \sum_{i=0}^{m-1} \frac{\gamma_{m,i}}{2k} \|u - U_i\|^2 \\ &= \arg \min_u E_1(u) + \sum_{i=0}^{m-1} \theta_{m,i} [L_2(u, U_i) + \frac{\Lambda}{2} \|u - U_i\|^2] + \frac{1}{2k} \sum_{j=m}^M \frac{\tilde{S}_{j,m}^2}{\tilde{S}_{j,j}} \left\| u - \sum_{i=0}^{m-1} \frac{\tilde{\gamma}_{j,i}}{\tilde{S}_{j,m}} U_i \right\|^2. \end{aligned}$$

Proof. As in the two step case the proof consists of expanding the norm squared terms and showing that all the quadratic and linear terms of u are equal. First, the expansion of $\sum_{i=0}^{m-1} \frac{\gamma_{m,i}}{2k} \|u - U_i\|^2$ is

$$\frac{\|u\|^2}{2k} \sum_{i=0}^{m-1} \gamma_{m,i} - \frac{1}{k} \langle u, \sum_{i=0}^{m-1} \gamma_{m,i} U_i \rangle + \text{terms that do not depend on } u. \quad (2.10)$$

Next, we will establish two identities to help us expand

$$\frac{1}{2k} \sum_{j=m}^M \frac{\tilde{S}_{j,m}^2}{\tilde{S}_{j,j}} \left\| u - \sum_{i=0}^{m-1} \frac{\tilde{\gamma}_{j,i}}{\tilde{S}_{j,m}} U_i \right\|^2.$$

First by rearranging (2.8),

$$\gamma_{m,i} - k\Lambda\theta_{m,i} = \sum_{j=m}^M \tilde{\gamma}_{j,i} \frac{\tilde{S}_{j,m}}{\tilde{S}_{j,j}}. \quad (2.11)$$

Next, an identity of $\tilde{S}_{m,m}$:

$$\begin{aligned}\tilde{S}_{m,m} &= \sum_{i=0}^{m-1} \tilde{\gamma}_{m,i} = \sum_{i=0}^{m-1} \left[\gamma_{m,i} - k\Lambda\theta_{m,i} - \sum_{j=m+1}^M \tilde{\gamma}_{j,i} \frac{\tilde{S}_{j,m}}{\tilde{S}_{j,j}} \right] \\ &= \sum_{i=0}^{m-1} \left[\gamma_{m,i} - k\Lambda\theta_{m,i} \right] - \sum_{j=m+1}^M \left[\sum_{i=0}^{m-1} \tilde{\gamma}_{j,i} \right] \frac{\tilde{S}_{j,m}}{\tilde{S}_{j,j}} \\ &= \sum_{i=0}^{m-1} \left[\gamma_{m,i} - k\Lambda\theta_{m,i} \right] - \sum_{j=m+1}^M \frac{\tilde{S}_{j,m}^2}{\tilde{S}_{j,j}}.\end{aligned}$$

We use this identity to establish the following:

$$\sum_{j=m}^M \frac{\tilde{S}_{j,m}^2}{\tilde{S}_{j,j}} = \tilde{S}_{m,m} + \sum_{j=m+1}^M \frac{\tilde{S}_{j,m}^2}{\tilde{S}_{j,j}} = \sum_{i=0}^{m-1} \left[\gamma_{m,i} - k\Lambda\theta_{m,i} \right] \quad (2.12)$$

Now we can calculate the expansion:

$$\begin{aligned}& \frac{1}{2k} \sum_{j=m}^M \frac{\tilde{S}_{j,m}^2}{\tilde{S}_{j,j}} \|u - \sum_{i=0}^{m-1} \frac{\tilde{\gamma}_{j,i}}{\tilde{S}_{j,m}} U_i\|^2 + \frac{1}{2} \sum_{i=0}^{m-1} \theta_{m,i} \Lambda \|u - U_i\|^2 \\ &= \frac{\|u\|^2}{2k} \sum_{j=m}^M \frac{\tilde{S}_{j,m}^2}{\tilde{S}_{j,j}} - \frac{1}{k} \langle u, \sum_{i=0}^{m-1} \sum_{j=m}^M \tilde{\gamma}_{j,i} \frac{\tilde{S}_{j,m}}{\tilde{S}_{j,j}} U_i \rangle + \frac{\Lambda}{2} \|u\|^2 \sum_{i=0}^{m-1} \theta_{m,i} - \Lambda \sum_{i=0}^{m-1} \langle u, \theta_{m,i} U_i \rangle \\ &\quad + \text{terms that do not depend on } u \\ &= \frac{\|u\|^2}{2k} \sum_{i=0}^{m-1} \gamma_{m,i} - \frac{1}{k} \langle u, \sum_{i=0}^{m-1} \gamma_{m,i} U_i \rangle + \text{terms that do not depend on } u.\end{aligned}$$

Where the last equality follows from (2.11) and (2.12). Since this expansion matches (2.10) up to a constant in u the proof is complete. $\square$

Lemma 2.3. *Let Λ and the auxiliary quantities $\tilde{S}_{j,m}$, $\tilde{\gamma}_{m,i}$ be given in theorem 2.1. Additionally, let $\tilde{S}_{m,m} > 0$ for $m = 1, \dots, M$. Then*

$$\begin{aligned}& E_1(U_m) + \sum_{i=0}^{m-1} \theta_{m,i} [L_2(U_m, U_i) + \frac{\Lambda}{2} \|U_m - U_i\|^2] + \frac{1}{2k} \sum_{j=m}^M \frac{\tilde{S}_{j,m}^2}{\tilde{S}_{j,j}} \left\| U_m - \sum_{i=0}^{m-1} \frac{\tilde{\gamma}_{j,i}}{\tilde{S}_{j,m}} U_i \right\|^2 \\ &\leq E_1(U_{m-1}) + \sum_{i=0}^{m-2} \theta_{m-1,i} [L_2(U_{m-1}, U_i) + \frac{\Lambda}{2} \|U_{m-1} - U_i\|^2] \\ &\quad + \frac{1}{2k} \sum_{j=m-1}^M \frac{\tilde{S}_{j,m-1}^2}{\tilde{S}_{j,j}} \left\| U_{m-1} - \sum_{i=0}^{m-2} \frac{\tilde{\gamma}_{j,i}}{\tilde{S}_{j,m-1}} U_i \right\|^2\end{aligned}$$

Proof. By (2.1) & lemma 2.2,

$$U_m = \arg \min_u E_1(u) + \sum_{i=0}^{m-1} \theta_{m,i} [L_2(u, U_i) + \frac{\Lambda}{2} \|u - U_i\|^2] + \frac{1}{2k} \sum_{j=m}^M \frac{\tilde{S}_{j,m}^2}{\tilde{S}_{j,j}} \|u - \sum_{i=0}^{m-1} \frac{\tilde{\gamma}_{j,i}}{\tilde{S}_{j,m}} U_i\|^2.$$

Since U_m is the minimizer of the above optimization problem

$$\begin{aligned}
& E_1(U_m) + \sum_{i=0}^{m-1} \theta_{m,i} [L_2(U_m, U_i) + \frac{\Lambda}{2} \|U_m - U_i\|^2] \\
& + \frac{1}{2k} \sum_{j=m}^M \frac{\tilde{S}_{j,m}^2}{\tilde{S}_{j,j}} \left\| U_m - \sum_{i=0}^{m-1} \frac{\tilde{\gamma}_{j,i}}{\tilde{S}_{j,m}} U_i \right\|^2 \\
& \leq E_1(U_{m-1}) + \theta_{m,m-1} E_2(U_{m-1}) + \sum_{i=0}^{m-2} \theta_{m,i} [L_2(U_{m-1}, U_i) + \frac{\Lambda}{2} \|U_{m-1} - U_i\|^2] \\
& + \frac{1}{2k} \sum_{j=m}^M \frac{\tilde{S}_{j,m}^2}{\tilde{S}_{j,j}} \left\| U_{m-1} - \sum_{i=0}^{m-1} \frac{\tilde{\gamma}_{j,i}}{\tilde{S}_{j,m}} U_i \right\|^2.
\end{aligned} \tag{2.13}$$

We give two inequalities to aid us in the proof. First, using the definition of the auxiliary variables, we can state an identity that will simplify (2.13). For $m > 1$ and $j \geq m$

$$\begin{aligned}
& \frac{\tilde{S}_{j,m}^2}{\tilde{S}_{j,j}} \left\| U_{m-1} - \sum_{i=0}^{m-1} \frac{\tilde{\gamma}_{j,i}}{\tilde{S}_{j,m}} U_i \right\|^2 = \frac{\tilde{S}_{j,m}^2}{\tilde{S}_{j,j}} \left\| U_{m-1} \left(1 - \frac{\tilde{\gamma}_{j,m-1}}{\tilde{S}_{j,m}} \right) - \sum_{i=0}^{m-2} \frac{\tilde{\gamma}_{j,i}}{\tilde{S}_{j,m}} U_i \right\|^2 \\
& = \frac{\tilde{S}_{j,m}^2}{\tilde{S}_{j,j}} \left\| U_{m-1} \left(\frac{\tilde{S}_{j,m-1}}{\tilde{S}_{j,m}} \right) - \sum_{i=0}^{m-2} \frac{\tilde{\gamma}_{j,i}}{\tilde{S}_{j,m}} U_i \right\|^2 = \frac{\tilde{S}_{j,m-1}^2}{\tilde{S}_{j,j}} \left\| U_{m-1} - \sum_{i=0}^{m-2} \frac{\tilde{\gamma}_{j,i}}{\tilde{S}_{j,m-1}} U_i \right\|^2.
\end{aligned} \tag{2.14}$$

Now since $\tilde{S}_{m-1,m-1} > 0$,

$$\frac{\tilde{S}_{m-1,m-1}^2}{\tilde{S}_{m-1,m-1}} \left\| U_{m-1} - \sum_{i=0}^{m-2} \frac{\tilde{\gamma}_{m-1,i}}{\tilde{S}_{m-1,m-1}} U_i \right\|^2 > 0. \tag{2.15}$$

Using (2.14) and (2.15) we have

$$\begin{aligned}
& \frac{1}{2k} \sum_{j=m}^M \frac{\tilde{S}_{j,m}^2}{\tilde{S}_{j,j}} \left\| U_{m-1} - \sum_{i=0}^{m-1} \frac{\tilde{\gamma}_{j,i}}{\tilde{S}_{j,m}} U_i \right\|^2 = \frac{1}{2k} \sum_{j=m}^M \frac{\tilde{S}_{j,m-1}^2}{\tilde{S}_{j,j}} \left\| U_{m-1} - \sum_{i=0}^{m-2} \frac{\tilde{\gamma}_{j,i}}{\tilde{S}_{j,m-1}} U_i \right\|^2 \\
& \leq \frac{1}{2k} \sum_{j=m-1}^M \frac{\tilde{S}_{j,m-1}^2}{\tilde{S}_{j,j}} \left\| U_{m-1} - \sum_{i=0}^{m-2} \frac{\tilde{\gamma}_{j,i}}{\tilde{S}_{j,m-1}} U_i \right\|^2.
\end{aligned} \tag{2.16}$$

Next, since $\sum_{i=1}^{m-1} \theta_{m,i} = 1$ for all m we have the equality

$$\theta_{m,m-1} = 1 - \sum_{i=0}^{m-2} \theta_{m,i} = \sum_{i=0}^{m-2} \theta_{m-1,i} - \sum_{i=0}^{m-2} \theta_{m,i}. \tag{2.17}$$

Using (2.5) and (2.17), we have our second inequality:

$$\begin{aligned}
& \theta_{m,m-1} E_2(U_{m-1}) + \sum_{i=0}^{m-2} \theta_{m,i} [L_2(U_{m-1}, U_i) + \frac{\Lambda}{2} \|U_{m-1} - U_i\|^2] \\
&= \sum_{i=0}^{m-2} (\theta_{m-1,i} - \theta_{m,i}) E_2(U_{m-1}) + \sum_{i=0}^{m-2} \theta_{m,i} [L_2(U_{m-1}, U_i) + \frac{\Lambda}{2} \|U_{m-1} - U_i\|^2] \\
&\leq \sum_{i=0}^{m-2} (\theta_{m-1,i} - \theta_{m,i}) [L_2(U_{m-1}, U_i) + \frac{\Lambda}{2} \|U_{m-1} - U_i\|^2] \\
&\quad + \sum_{i=0}^{m-2} \theta_{m,i} [L_2(U_{m-1}, U_i) + \frac{\Lambda}{2} \|U_{m-1} - U_i\|^2] \\
&= \sum_{i=0}^{m-2} \theta_{m-1,i} [L_2(U_{m-1}, U_i) + \frac{\Lambda}{2} \|U_{m-1} - U_i\|^2].
\end{aligned} \tag{2.18}$$

Using inequalities (2.16) and (2.18), we have that (2.13) is less than or equal to

$$\begin{aligned}
& E_1(U_{m-1}) + \sum_{i=0}^{m-2} \theta_{m-1,i} [L_2(U_{m-1}, U_i) + \frac{\Lambda}{2} \|U_{m-1} - U_i\|^2] \\
&\quad + \frac{1}{2k} \sum_{j=m-1}^M \frac{\tilde{S}_{j,m-1}^2}{\tilde{S}_{j,j}} \left\| U_{m-1} - \sum_{i=0}^{m-2} \frac{\tilde{\gamma}_{j,i}}{\tilde{S}_{j,m-1}} U_i \right\|^2,
\end{aligned}$$

concluding the proof. $\square$

Proof. (of theorem) The main idea of the proof is to use lemma 2.3 repeatedly to relate the energy of $E(u_{n+1})$ to $E(u_n)$. First, by (2.5) and our assumption that $\tilde{S}_{M,M} > 0$

$$\begin{aligned}
E(u_{n+1}) &= E_1(U_M) + E_2(U_M) \\
&\leq E_1(U_M) + \sum_{i=0}^{M-1} \theta_{M,i} [L_2(U_M, U_i) + \frac{\Lambda}{2} \|U_M - U_i\|^2] \\
&\quad + \frac{1}{2k} \frac{\tilde{S}_{M,M}^2}{\tilde{S}_{M,M}} \|U_M - \sum_{i=0}^{M-1} \frac{\tilde{\gamma}_{M,i}}{\tilde{S}_{M,M}} U_i\|^2.
\end{aligned}$$

By using the lemma 2.3 repeatedly we have

$$\begin{aligned}
& E_1(U_M) + \sum_{i=0}^{M-1} \theta_{M,i} [L_2(U_M, U_i) + \frac{\Lambda}{2} \|U_M - U_i\|^2] + \frac{1}{2k} \frac{\tilde{S}_{M,M}^2}{\tilde{S}_{M,M}} \|U_M - \sum_{i=0}^{M-1} \frac{\tilde{\gamma}_{M,i}}{\tilde{S}_{M,M}} U_i\|^2 \\
& \leq E_1(U_{M-1}) + \sum_{i=0}^{M-2} \theta_{M-1,i} [L_2(U_{M-1}, U_i) + \frac{\Lambda}{2} \|U_{M-1} - U_i\|^2] \\
& \quad + \frac{1}{2k} \sum_{j=M-1}^M \frac{\tilde{S}_{j,M-1}^2}{\tilde{S}_{j,j}} \left\| U_{M-1} - \sum_{i=0}^{M-2} \frac{\tilde{\gamma}_{j,i}}{\tilde{S}_{j,M-1}} U_i \right\|^2 \\
& \quad \vdots \\
& \leq E_1(U_1) + L_2(U_1, U_0) + \frac{\Lambda}{2} \|U_1 - U_0\|^2 + \frac{1}{2k} \sum_{j=1}^M \frac{\tilde{S}_{j,1}^2}{\tilde{S}_{j,j}} \|U_1 - \frac{\tilde{\gamma}_{j,0}}{\tilde{S}_{j,1}} U_0\|^2.
\end{aligned}$$

By (2.1) and lemma 2.2,

$$U_1 = \arg \min_u E_1(u) + L_2(u, U_0) + \frac{\Lambda}{2} \|u - U_0\|^2 + \frac{1}{2k} \sum_{j=1}^M \frac{\tilde{S}_{j,1}^2}{\tilde{S}_{j,j}} \|u - \frac{\tilde{\gamma}_{j,0}}{\tilde{S}_{j,1}} U_0\|^2,$$

so

$$\begin{aligned}
& E_1(U_1) + L_2(U_1, U_0) + \frac{\Lambda}{2} \|U_1 - U_0\|^2 + \frac{1}{2k} \sum_{j=1}^M \frac{\tilde{S}_{j,1}^2}{\tilde{S}_{j,j}} \|U_1 - \frac{\tilde{\gamma}_{j,0}}{\tilde{S}_{j,1}} U_0\|^2 \\
& \leq E_1(U_0) + E_2(U_0) + \frac{\Lambda}{2} \|U_0 - U_0\|^2 + \frac{1}{2k} \sum_{j=1}^M \frac{\tilde{S}_{j,1}^2}{\tilde{S}_{j,j}} \|U_0 - U_0\|^2 \\
& = E(u_n),
\end{aligned}$$

completing the proof of the theorem. $\square$

Remark 2.4. In the above proof, we assume that $E_2(u)$ is two times differentiable. This assumption can be dropped if we replace $L_2(u, p)$ with another approximation $A_2(u, p)$ that has the properties $A_2(u, u) = E_2(u)$ and for some choice Λ , $E_2(u) \leq A_2(u, p) + \frac{\Lambda}{2} \|u - p\|^2$ for all u and p .

3. Examples of the New Schemes for Gradient Flows

In this section, we give examples of *high order* semi-implicit schemes for gradient flows, for any desired choice of implicit and explicit terms E_1 and E_2 , that are energy stable under the conditions of theorem 2.1. First, we give the conditions on $\gamma_{m,i}$ and $\theta_{m,i}$ in scheme (2.1) to ensure high order consistency

with the abstract evolution law (1.1). Recall that $U_0 = u_n$. From (2.1), each stage U_m satisfies the Euler-Lagrange equation:

$$\left[\sum_{i=0}^{m-1} \gamma_{m,i} \right] U_m + k \nabla_H E_1(U_m) = - \sum_{i=0}^{m-1} k \theta_{m,i} \nabla_H E_2(U_i) + \sum_{i=0}^{m-1} \gamma_{m,i} U_i. \quad (3.1)$$

Equation (3.1) is equivalent to the form more often seen for ARK IMEX methods:

$$U_m = U_0 - k \sum_{i=1}^m \alpha_{m,i} \nabla_H E_1(U_i) - k \sum_{i=1}^{m-1} \tilde{\alpha}_{m,i} \nabla_H E_2(U_i) \quad (3.2)$$

where $\alpha_{m,i}$ and $\tilde{\alpha}_{m,i}$ depend on $\gamma_{m,i}$ and $\theta_{m,i}$. The consistency equations for ARK IMEX methods have been previously worked out [3, 29, 30, 31]. As such, we will state without proof the conditions required to achieve various orders of accuracy in terms of γ and θ :

Claim 3.1. *Let U_i be given in (2.1). The Taylor expansion of U_i at each stage has the form:*

$$\begin{aligned} U_i = & U_0 - k \beta_{1,i} DE(U_0) + k^2 [\beta_{2,i} D^2 E_1(U_0) DE(U_0) + \beta_{3,i} D^2 E_2(U_0) DE(U_0)] \\ & - k^3 [\beta_{4,i} D^2 E_1(U_0) (D^2 E_1(U_0) (DE(U_0))) + \beta_{5,i} D^2 E_1(U_0) (D^2 E_2(U_0) (DE(U_0))) \\ & + \beta_{6,i} D^2 E_2(U_0) (D^2 E_1(U_0) (DE(U_0))) + \beta_{7,i} D^2 E_2(U_0) (D^2 E_2(U_0) (DE(U_0))) \\ & + \beta_{8,i} D^3 E_1(U_0) (DE(U_0), DE(U_0)) + \beta_{9,i} D^3 E_2(U_0) (DE(U_0), DE(U_0))] + h.o.t. \end{aligned} \quad (3.3)$$

where for $l \in \{1, 2, 3, \dots\}$, $D^l E(u) : H^l \rightarrow \mathbb{R}$ denotes the multilinear form given by

$$D^l E(u)(v_1, \dots, v_l) = \frac{\partial^l}{\partial s_1 \dots \partial s_l} E(u + s_1 v_1 + s_2 v_2 + \dots + s_l v_l) \Big|_{s_1 = s_2 = \dots = s_l = 0}$$

so that the linear functional $D^l E(u)(v_1, v_2, \dots, v_{l-1}, \cdot) : H \rightarrow \mathbb{R}$ may be identified with an element of H , and so on. The coefficients of (3.3) obey the following recursive relations:

$$\begin{aligned}
\beta_{1,0} &= \beta_{2,0} = \dots = \beta_{9,0} = 0, \\
\beta_{1,m} &= \frac{1}{S_m} \left[1 + \sum_{i=1}^{m-1} \gamma_{m,i} \beta_{1,i} \right], \\
\beta_{2,m} &= \frac{1}{S_m} \left[\beta_{1,m} + \sum_{i=1}^{m-1} \gamma_{m,i} \beta_{2,i} \right], \\
\beta_{3,m} &= \frac{1}{S_m} \left[\sum_{i=0}^{m-1} \theta_{m,i} \beta_{1,i} + \sum_{i=1}^{m-1} \gamma_{m,i} \beta_{3,i} \right], \\
\beta_{4,m} &= \frac{1}{S_m} \left[\beta_{2,m} + \sum_{i=1}^{m-1} \gamma_{m,i} \beta_{4,i} \right], \\
\beta_{5,m} &= \frac{1}{S_m} \left[\beta_{3,m} + \sum_{i=1}^{m-1} \gamma_{m,i} \beta_{5,i} \right], \\
\beta_{6,m} &= \frac{1}{S_m} \left[\sum_{i=0}^{m-1} \theta_{m,i} \beta_{2,i} + \sum_{i=1}^{m-1} \gamma_{m,i} \beta_{6,i} \right], \\
\beta_{7,m} &= \frac{1}{S_m} \left[\sum_{i=0}^{m-1} \theta_{m,i} \beta_{3,i} + \sum_{i=1}^{m-1} \gamma_{m,i} \beta_{7,i} \right], \\
\beta_{8,m} &= \frac{1}{S_m} \left[\frac{\beta_{1,m}^2}{2} + \sum_{i=1}^{m-1} \gamma_{m,i} \beta_{8,i} \right], \\
\beta_{9,m} &= \frac{1}{S_m} \left[\frac{1}{2} \sum_{i=0}^{m-1} \theta_{m,i} \beta_{1,i}^2 + \sum_{i=1}^{m-1} \gamma_{m,i} \beta_{9,i} \right],
\end{aligned} \tag{3.4}$$

with $S_m = \sum_{i=0}^{m-1} \gamma_{m,i}$. Furthermore, the following conditions for $u_{n+1} = U_M$ in scheme (2.1) are necessary and sufficient for various orders of accuracy:

<u>First Order:</u>	<u>Second Order:</u>	<u>Third Order:</u>	
$\beta_{1,M} = 1$	$\beta_{1,M} = 1$	$\beta_{1,M} = 1$	
	$\beta_{2,M} = 1/2$	$\beta_{2,M} = 1/2$	(3.5)
	$\beta_{3,M} = 1/2$	$\beta_{3,M} = 1/2$	
		$\beta_{4,M} = \beta_{5,M} = \dots = \beta_{9,M} = 1/6.$	

Now, we give a second order and a third order example of method (2.1). All

of those coefficients in our examples are rational values, albeit with long representations. For simplicity, we give rounded decimal values here. The coefficients to machine precision as well as code to verify theorem 2.1 and claim 3.1 can be found at <https://github.com/AZaitzeff/Sigradflow>. However, The examples we give are not unique by any means. We begin with a five step method that is second order accurate:

$$\begin{aligned} \theta &\approx \begin{pmatrix} 1. & 0 & 0 & 0 & 0 \\ 0.009 & 0.991 & 0 & 0 & 0 \\ 0.009 & 0.991 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1. & 0 \\ 0 & 0 & 0 & 1. & 0 \end{pmatrix}, \\ \gamma &\approx \begin{pmatrix} 8.841 & 0 & 0 & 0 & 0 \\ -0.925 & 5.360 & 0 & 0 & 0 \\ -4.443 & 6.041 & 0.950 & 0 & 0 \\ -3.288 & 5.895 & -0.351 & 0.172 & 0 \\ -3.895 & -0.335 & 4.964 & -1.722 & 7.684 \end{pmatrix} \end{aligned} \tag{3.6}$$

which is stable for $k\Lambda \leq 3/872$. Next we have a thirteen step method that

is third order accurate:

$$\begin{aligned} \theta &\approx \begin{pmatrix} 1. & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.049 & 0.951 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.024 & 0.075 & 0.901 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.017 & 0.042 & 0.113 & 0.829 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.012 & 0.029 & 0.071 & 0.386 & 0.501 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.01 & 0.023 & 0.06 & 0.366 & 0.457 & 0.085 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.007 & 0.018 & 0.05 & 0.351 & 0.437 & 0.06 & 0.076 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.003 & 0.005 & 0.006 & 0.008 & 0.009 & 0.011 & 0.028 & 0.929 & 0 & 0 & 0 & 0 & 0 \\ 0.002 & 0.002 & 0.002 & 0.002 & 0.003 & 0.004 & 0.009 & 0.029 & 0.948 & 0 & 0 & 0 & 0 \\ 0 & 0.001 & 0.001 & 0.001 & 0.001 & 0.002 & 0.004 & 0.007 & 0.011 & 0.971 & 0 & 0 & 0 \\ 0 & 0 & 0.001 & 0.001 & 0.001 & 0.001 & 0.003 & 0.005 & 0.008 & 0.912 & 0.069 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.001 & 0.002 & 0.003 & 0.005 & 0.107 & 0.025 & 0.857 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.001 & 0.001 & 0.002 & 0.013 & 0.007 & 0.018 & 0.958 \end{pmatrix}, \\ \gamma &\approx \begin{pmatrix} 11. & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2.1 & 15.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.4 & 1.6 & 17. & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.2 & 1.6 & -2.4 & 18.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.3 & -8.5 & 3. & 9.6 & 7.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1.4 & -5.9 & -0.1 & 2. & 8. & 4.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -4. & -0.5 & -0.4 & -1.8 & 5.1 & 6.8 & 0.9 & 0 & 0 & 0 & 0 & 0 & 0 \\ -9.2 & 4.8 & 2.7 & -3.2 & 2.5 & 6.2 & 2.5 & 4.6 & 0 & 0 & 0 & 0 & 0 \\ -1.7 & -3.6 & -0.1 & 1.3 & 5.7 & 3.4 & -0.8 & -0.8 & 0.4 & 0 & 0 & 0 & 0 \\ -2.7 & -3.5 & 0.6 & 1.4 & 6.1 & 3.5 & -0.7 & -0.2 & -0.4 & 0.5 & 0 & 0 & 0 \\ 5.9 & -4.8 & -5.1 & -3.1 & 3.4 & 6.6 & -0.7 & -5.2 & 4.9 & -0.8 & 8.2 & 0 & 0 \\ 7.1 & 0.9 & -3.1 & -2.7 & -5.8 & -1.9 & 0.6 & -3.4 & 4.3 & -1.3 & 9.2 & 9.1 & 0 \\ 3.8 & 1.9 & 2.7 & 2.1 & -7.5 & -10.6 & -1.2 & 2. & 0.7 & -0.2 & -0.2 & 9.5 & 12.8 \end{pmatrix} \end{aligned} \tag{3.7}$$

which is stable if $k\Lambda \leq 18/28567$.

The energy stability and consistency of the schemes discussed above can often be readily harnessed to establish their convergence. For instance, this can be accomplished with quite familiar arguments in the context of Allen-Cahn and Cahn-Hilliard equations. Demonstrations of this, at the expected rate of convergence, is an important supplement of the present work that will be taken up in a subsequent contribution.

In the following section, we consider methods for (1.2), when the inner product changes with the solution.

4. Schemes for Solving Gradient Flows with Solution Dependent Inner Product

Now we move on to the problem of simulating flow (1.2),

$$u' = -\mathcal{L}(u)\nabla_H E(u).$$

We consider the case where $\mathcal{L}(u)$ is strictly positive definite. Our approach will be as follows:

1. Generate a u_* from u_n .
2. Construct $\mathcal{L}(u_*)$.
3. Use the algorithm (2.1) with norm $\|\cdot\|_{\mathcal{L}^{-1}(u_*)}^2 = \langle \cdot, \mathcal{L}^{-1}(u_*) \cdot \rangle$ to generate u_{n+1} .

One advantage to constructing $\mathcal{L}(u_*)$ and then using it in (2.1) is that theorem 2.1 immediately gives conditional energy stability for coefficients such as (3.6) or (3.7). Thus, we only need to consider what choice of u_* will give our algorithm the desired level of accuracy. Now at every step we are solving

$$\left[\sum_{i=0}^{m-1} \gamma_{m,i} \right] U_m + k \mathcal{L}(u_*) \nabla_H E_1(U_m) = -k \mathcal{L}(u_*) \sum_{i=0}^{m-1} \theta_{m,i} \nabla_H E_2(U_i) + \sum_{i=0}^{m-1} \gamma_{m,i} U_i. \quad (4.1)$$

We will set up the consistency equations for (1.2). Let $u_n = u(t_0)$. For convenience, denote $\mathcal{L}(u_n)$ as $\mathcal{L}_n$ and $E(u_n)$ as E_n . We begin with the exact solution starting from $u(t_0)$:

$$\begin{cases} u_t = -\mathcal{L}(u) \nabla E(u) & t > t_0 \\ u(t_0) = U_0 \end{cases}$$

By Taylor expanding around t_0 we find

$$u(k + t_0) = u(t_0) + k u_t(t_0) + \frac{1}{2} k^2 u_{tt}(t_0) + \frac{1}{6} k^3 u_{ttt}(t_0) \quad (4.2)$$

where the higher derivatives in time are found using (1.2):

$$\begin{aligned} u_t(t_0) &= -\mathcal{L}_n D E_n \\ u_{tt}(t_0) &= D \mathcal{L}_n (\mathcal{L}_n D E_n) D E_n + \mathcal{L}_n D^2 E_n (\mathcal{L}_n D E_n) \\ u_{ttt}(t_0) &= -D \mathcal{L}_n (D \mathcal{L}_n (\mathcal{L}_n D E_n) D E_n) D E_n - D^2 \mathcal{L}_n (\mathcal{L}_n D E_n, \mathcal{L}_n D E_n) D E_n \\ &\quad - D \mathcal{L}_n (\mathcal{L}_n (D^2 E_n (\mathcal{L}_n D E_n))) D E_n - 2 D \mathcal{L}_n (\mathcal{L}_n D E_n) D^2 E_n (\mathcal{L}_n D E_n) \\ &\quad - \mathcal{L}_n D^2 E_n (D \mathcal{L}_n (\mathcal{L}_n D E_n)) D E_n - \mathcal{L}_n D^2 E_n (\mathcal{L}_n D^2 E_n (\mathcal{L}_n D E_n)) \\ &\quad - \mathcal{L}_n D^3 E_n (\mathcal{L}_n D E_n, \mathcal{L}_n D E_n) \end{aligned}$$

where for $l \in \{1, 2, 3, \dots\}$, $D^l L(u) : H^l \rightarrow H$ denotes the multilinear form given by

$$D^l \mathcal{L}(u)(v_1, \dots, v_l) = \frac{\partial^l}{\partial s_1 \dots \partial s_l} \mathcal{L}(u + s_1 v_1 + s_2 v_2 + \dots + s_l v_l) \Big|_{s_1=s_2=\dots=s_l=0}$$

so that $D^l \mathcal{L}(u)(v_1, v_2, \dots, v_l)$ is a linear operator from H to H .

In the next two subsections, we provide second and third order examples and accompanying consistency calculations. Both of these examples also have the property that $E(u_*) \leq E(u_n)$.

4.1. Second Order Method

Algorithm 1 A second order method for solving gradient flows with solution dependent inner product

Fix a time step size $k > 0$. Set $u_n = u_0$. To obtain u_{n+1} from u_n , carry out the following steps:

1. Find u_* by solving $u_* + \frac{1}{2}k\mathcal{L}_n \nabla E_1(u_*) = u_n - \frac{1}{2}k\mathcal{L}_n \nabla E_2(u_n)$.
 2. Find u_{n+1} using (4.1) with coefficients (3.6) and u_* in $\mathcal{L}(u_*)$ given by step (1) of this algorithm.
-

Our second order algorithm is laid out in algorithm 1. Now we will prove that it is indeed second order. First, the expansion of u_* is

$$u_* = u_n - \frac{1}{2}k\mathcal{L}_n DE_n + O(k^2). \quad (4.3)$$

We use (3.4) to get an expansion of u_{n+1} :

$$u_{n+1} = u_n - k\mathcal{L}(u_*)DE_n + \frac{1}{2}k^2\mathcal{L}(u_*)D^2E_n(\mathcal{L}(u_*)DE_n) + O(k^3). \quad (4.4)$$

Now, expand u_* around u_n in (4.4):

$$\begin{aligned} u_{n+1} &= u_n - k\mathcal{L}_n DE_n - kD\mathcal{L}_n(u_* - u_n)DE_n \\ &\quad + \frac{1}{2}k^2\mathcal{L}_n D^2E_n(\mathcal{L}_n DE_n) + O(k^3) \\ &= u_n - k\mathcal{L}_n DE_n + \frac{1}{2}k^2D\mathcal{L}_n(\mathcal{L}_n DE_n)DE_n \\ &\quad + \frac{1}{2}k^2\mathcal{L}_n D^2E_n(\mathcal{L}_n DE_n) + O(k^3). \end{aligned}$$

The Taylor expansion of u_{n+1} matches (4.2) to second order.

4.2. Third Order Method

Algorithm 2 A third order method for solving gradient flows with solution dependent inner product

Fix a time step size $k > 0$ and set $u_n = u_0$. For convenience, we will denote $D^2\mathcal{L}(u_*)(\mathcal{L}(u_*)\nabla E(u_*), \mathcal{L}(u_*)\nabla E(u_*))$ as $D^2\mathcal{L}(u_*)$. Additionally, let $MS(\tilde{k}, \mathcal{L}(u_*), \tilde{u}, \gamma, \theta)$ denote U_M obtained from the multistage algorithm

$$\left[\sum_{i=0}^{m-1} \gamma_{m,i} \right] U_m + \tilde{k} \mathcal{L}(u_*) \nabla_H E_1(U_m) = -\tilde{k} \mathcal{L}(u_*) \sum_{i=0}^{m-1} \theta_{m,i} \nabla_H E_2(U_i) + \sum_{i=0}^{m-1} \gamma_{m,i} U_i.$$

with $U_0 = \tilde{u}$. To obtain u_{n+1} from u_n , carry out the following steps:

1. Let γ and θ be given by (4.5). Set $u_{*1} = MS\left(\frac{1}{6}k, \mathcal{L}(u_n), u_n, \gamma, \theta\right)$.
2. Let γ and θ be given by (3.7). Set

$$\bar{u} = MS\left(\frac{1}{2}k, \mathcal{L}(u_{*1}) - \frac{1}{72}k^2 D^2\mathcal{L}(u_{*1}), u_n, \gamma, \theta\right).$$

3. Let $\gamma = \theta = (1)$. Set $u_{*2,1} = MS\left(\frac{2}{5}k, \mathcal{L}(u_n), u_n, \gamma, \theta\right)$.
4. Let γ and θ be given by (4.6). Set $u_{*2,2} = MS\left(\frac{5}{6}k, \mathcal{L}(u_{*2,1}), u_n, \gamma, \theta\right)$.
5. Let γ and θ be given by (3.7). Then

$$u_{n+1} = MS\left(\frac{1}{2}k, \mathcal{L}(u_{*2,2}) - \frac{1}{72}k^2 D^2\mathcal{L}(u_{*2,2}), \bar{u}, \gamma, \theta\right).$$

Now we present our third order algorithm for solving (1.2). It requires the use of two new sets of coefficients,

$$\begin{aligned} \theta &\approx \begin{pmatrix} 1. & 0 & 0. \\ -0.667 & 0.333 & 0 \\ 0 & 0 & 1.000 \end{pmatrix}, \\ \gamma &\approx \begin{pmatrix} 1.833 & 0 & 0. \\ 0.556 & 0.667 & 0 \\ 1.030 & -0.026 & 0.159 \end{pmatrix} \end{aligned} \tag{4.5}$$

and

$$\theta \approx \begin{pmatrix} 1. & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.708 & 0.292 & 0 & 0 & 0 & 0 & 0 \\ 0.013 & 0.018 & 0.969 & 0 & 0 & 0 & 0 \\ 0.008 & 0.012 & 0.867 & 0.113 & 0 & 0 & 0 \\ 0.006 & 0.009 & 0.206 & 0.056 & 0.724 & 0 & 0 \\ 0 & 0.005 & 0.05 & 0.025 & 0.053 & 0.867 & 0 \\ 0 & 0 & 0.015 & 0.009 & 0.015 & 0.04 & 0.920 \end{pmatrix}, \quad (4.6)$$

$$\gamma \approx \begin{pmatrix} 7.727 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.594 & 2.241 & 0 & 0 & 0 & 0 & 0 \\ 3.056 & -0.455 & 0.636 & 0 & 0 & 0 & 0 \\ -1.571 & 5.091 & -1.063 & 2.786 & 0 & 0 & 0 \\ -3.714 & 3.1 & -1.267 & 1.545 & 9.655 & 0 & 0 \\ -6.923 & 5.1 & -2.056 & 3.471 & 4.571 & 4.033 & 0 \\ -2.467 & -2.1 & 0.009 & -0.182 & 0.660 & 7.224 & 9.428 \end{pmatrix},$$

to achieve particular Taylor expansions as we explain later in the section. The values of (4.5) and (4.6) to machine precision can be found at <https://github.com/AZaitzeff/SIgradflow>.

Algorithm 2 details our third order version for solving gradient flows with solution dependent inner product. The method adds another condition for stability to hold, namely:

$$\mathcal{L}(u) - \frac{1}{72}k^2 D^2 \mathcal{L}(u)(w, w) \quad (4.7)$$

needs to be positive definite for all u and w . Now we will prove that algorithm 2 produces a third order approximation.

By applying (3.4), the coefficients (4.5) give the following expansion for u_{*1} :

$$u_{*1} = u_n - \frac{1}{6}k\mathcal{L}_n D E_n + \frac{1}{36}k^2 \mathcal{L}_n D^2 E_n (\mathcal{L}_n D E_n) + O(k^3) \quad (4.8)$$

Now we can expand $\bar{u}$ by using (3.4) and expanding u_{*1} around u_n

$$\begin{aligned}
\bar{u} &= u_n - \frac{1}{2}k\mathcal{L}(u_{*1})DE_n + \frac{1}{8}k^2\mathcal{L}(u_{*1})D^2E_n(\mathcal{L}(u_{*1})DE_n) \\
&\quad - \frac{1}{48}k^3\mathcal{L}(u_{*1})D^2E_n(\mathcal{L}(u_{*1})D^2E_n(\mathcal{L}(u_{*1})DE_n)) \\
&\quad - \frac{1}{48}k^3\mathcal{L}(u_{*1})D^3E_n(\mathcal{L}(u_{*1})DE_n, \mathcal{L}(u_{*1})DE_n) \\
&\quad + \frac{1}{144}k^3D^2\mathcal{L}(u_{*1})(\mathcal{L}(u_{*1})DE(u_{*1}), \mathcal{L}(u_{*1})DE(u_{*1}))DE_n + O(k^4) \\
&= u_n - \frac{1}{2}k\mathcal{L}_nDE_n + \frac{1}{12}k^2D\mathcal{L}_n(\mathcal{L}_nDE_n)DE_n + \frac{1}{8}k^2\mathcal{L}_nD^2E_n(\mathcal{L}_nDE_n) \\
&\quad - \frac{1}{72}k^3D\mathcal{L}_n(\mathcal{L}_nD^2E_n(\mathcal{L}_nDE_n))DE_n \\
&\quad - \frac{1}{48}k^3\mathcal{L}_nD^2E_n(D\mathcal{L}_n(\mathcal{L}_nDE_n)DE_n) \\
&\quad - \frac{1}{48}k^3D\mathcal{L}_n(\mathcal{L}_nDE_n)D^2E_n(\mathcal{L}_nDE_n) \\
&\quad - \frac{1}{48}k^3\mathcal{L}_nD^2E_n(\mathcal{L}_nD^2E_n(\mathcal{L}_nDE_n)) \\
&\quad - \frac{1}{48}k^3\mathcal{L}_nD^3E_n(\mathcal{L}_nDE_n, \mathcal{L}_nDE_n) + O(k^4).
\end{aligned} \tag{4.9}$$

Now we will apply the same steps to derive the expansions of $u_{*2,1}$,

$$u_{*2,1} = u_n - \frac{2}{5}k\mathcal{L}_nDE_n + O(k^2), \tag{4.10}$$

and $u_{*2,2}$,

$$\begin{aligned}
u_{*2,2} &= u_n - \frac{5}{6}k\mathcal{L}(u_{*2,1})DE_n + \frac{11}{36}k^2\mathcal{L}(u_{*2,1})D^2E_n(\mathcal{L}(u_{*2,1})DE_n) + O(k^3) \\
&= u_n - \frac{5}{6}k\mathcal{L}_nDE_n \\
&\quad + \frac{1}{3}k^2D\mathcal{L}_n(\mathcal{L}_nDE_n)DE_n + \frac{11}{36}k^2\mathcal{L}_nD^2E_n(\mathcal{L}_nDE_n) + O(k^3).
\end{aligned} \tag{4.11}$$

Finally, we can find the expansion of u_{n+1} . We will first apply (3.4) around $\bar{u}$,

$$\begin{aligned}
u_{n+1} &= \bar{u} - \frac{1}{2}k\mathcal{L}(u_{*2,2})DE(\bar{u}) + \frac{1}{8}k^2\mathcal{L}(u_{*2,2})D^2E(\bar{u})(\mathcal{L}(u_{*2,2})DE(\bar{u})) \\
&\quad - \frac{1}{48}k^3\mathcal{L}(u_{*2,2})D^2E(\bar{u})(\mathcal{L}(u_{*2,2})D^2E(\bar{u})(\mathcal{L}(u_{*2,2})DE(\bar{u}))) \\
&\quad - \frac{1}{48}k^3\mathcal{L}(u_{*2,2})D^3E(\bar{u})(\mathcal{L}(u_{*2,2})DE(\bar{u}), \mathcal{L}(u_{*2,2})DE(\bar{u})) \\
&\quad + \frac{1}{144}k^3D^2\mathcal{L}(u_{*2,2})(\mathcal{L}(u_{*2,2})DE(u_{*2,2}), \mathcal{L}(u_{*2,2})DE(u_{*2,2}))DE(\bar{u}) + O(k^4),
\end{aligned}$$

expand $u_{*,2,2}$,

$$\begin{aligned}
u_{n+1} = & \bar{u} - \frac{1}{2}k\mathcal{L}_n DE(\bar{u}) + \frac{5}{12}k^2 D\mathcal{L}_n(\mathcal{L}_n DE_n)DE(\bar{u}) + \frac{1}{8}k^2 \mathcal{L}_n D^2 E(\bar{u})(\mathcal{L}_n DE(\bar{u})) \\
& - \frac{1}{6}k^3 D\mathcal{L}_n(D\mathcal{L}_n(\mathcal{L}_n DE_n)DE_n)DE(\bar{u}) \\
& - \frac{1}{6}k^3 D^2 \mathcal{L}_n(\mathcal{L}_n DE_n, \mathcal{L}_n DE_n)DE(\bar{u}) \\
& - \frac{11}{72}k^3 D\mathcal{L}_n(\mathcal{L}_n D^2 E_n(\mathcal{L}_n DE_n))DE(\bar{u}) \\
& - \frac{5}{48}k^3 D\mathcal{L}_n(\mathcal{L}_n DE(\bar{u}))D^2 E(\bar{u})(\mathcal{L}_n DE(\bar{u})) \\
& - \frac{5}{48}k^3 \mathcal{L}_n D^2 E(\bar{u})(D\mathcal{L}_n(\mathcal{L}_n DE(\bar{u}))DE(\bar{u})) \\
& - \frac{1}{48}k^3 \mathcal{L}_n D^2 E(\bar{u})(\mathcal{L}_n D^2 E(\bar{u})(\mathcal{L}_n DE(\bar{u}))) \\
& - \frac{1}{48}k^3 \mathcal{L}_n D^3 E(\bar{u})(\mathcal{L}_n DE(\bar{u}), \mathcal{L}_n DE(\bar{u})) + O(k^4),
\end{aligned}$$

then expand $\bar{u}$ around u_n :

$$\begin{aligned}
u_{n+1} = & u_n - k\mathcal{L}_n DE_n + \frac{1}{2}k^2 D\mathcal{L}_n(\mathcal{L}_n DE_n)DE_n + \frac{1}{2}k^2 \mathcal{L}_n D^2 E_n(\mathcal{L}_n DE_n) \\
& - \frac{1}{6}k^3 D\mathcal{L}_n(D\mathcal{L}_n(\mathcal{L}_n DE_n)DE_n)DE_n \\
& - \frac{1}{6}k^3 D^2 \mathcal{L}_n(\mathcal{L}_n DE_n, \mathcal{L}_n DE_n)DE_n \\
& - \frac{1}{6}k^3 D\mathcal{L}_n(\mathcal{L}_n D^2 E_n(\mathcal{L}_n DE_n))DE_n \\
& - \frac{1}{3}k^3 D\mathcal{L}_n(\mathcal{L}_n DE_n)D^2 E_n(\mathcal{L}_n DE_n) \\
& - \frac{1}{6}k^3 \mathcal{L}_n D^2 E_n(D\mathcal{L}_n(\mathcal{L}_n DE_n)DE_n) \\
& - \frac{1}{6}k^3 \mathcal{L}_n D^2 E_n(\mathcal{L}_n D^2 E_n(\mathcal{L}_n DE_n)) \\
& - \frac{1}{6}k^3 \mathcal{L}_n D^3 E_n(\mathcal{L}_n DE_n, \mathcal{L}_n DE_n) + O(k^4).
\end{aligned}$$

The Taylor expansion of u_{n+1} matches (4.2) to third order. As long as (4.7) holds,

$$E(u_{n+1}) \leq E(\bar{u}) \leq E(u_n)$$

by theorem 2.1.

Remark 4.1. In algorithm 1 and algorithm 2, we can instead handle $E(u)$ fully implicitly as the authors do in [1]. We give these fully implicit versions of the algorithms in Appendix A.

5. Numerical Examples

In this section, we will apply the second and third order accurate conditionally stable schemes to a variety of gradient flows, some with fixed inner product and some with solution dependent inner product. Careful numerical convergence studies are presented in each case to verify the anticipated convergence rates of previous sections.

5.1. Gradient Flows with Fixed Inner Product

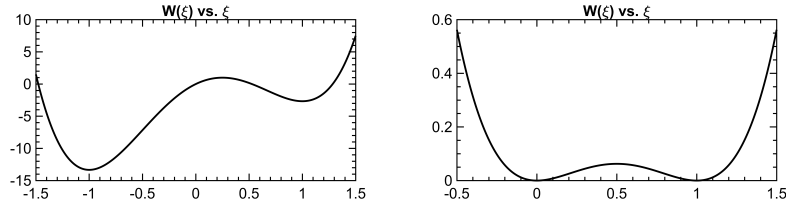


Figure 1: The double well potentials used in the Allen-Cahn (5.1) and Cahn-Hilliard (5.3) equations: One with unequal depth wells and the other with equal depth wells.

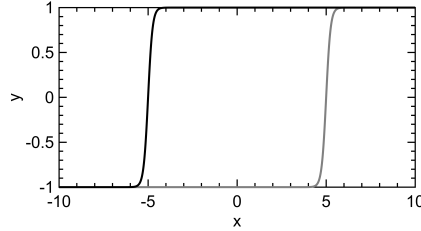


Figure 2: The initial condition (black) and the solution at final time (gray) in the numerical convergence study on the 1D Allen-Cahn equation (5.1) with a potential that has unequal depth wells.

We start with the Allen-Cahn equation

$$u_t = \Delta u - W'(u) \quad (5.1)$$

Number of time steps	2^9	2^{10}	2^{11}	2^{12}	2^{13}
L^2 error (2nd order)	2.08e-01	5.96e-02	1.61e-02	4.22e-03	1.08e-03
Order	-	1.81	1.89	1.94	1.97
L^2 error (3rd order)	2.06e-03	3.26e-04	4.68e-05	6.32e-06	8.33e-07
Order	-	2.66	2.80	2.89	2.92

Table 1: The new second (3.6) and third (3.7) order accurate, conditionally stable schemes (2.1) on the one-dimensional Allen-Cahn equation (5.1) with a traveling wave solution.

where $W : \mathbb{R} \rightarrow \mathbb{R}$ is a double-well potential. This corresponds to gradient flow for the energy

$$E(u) = \int \frac{1}{2} \|\nabla u\|^2 + W(u) dx \quad (5.2)$$

with respect to the L^2 inner product.

First, we consider equation (5.1) in one space dimension, with the potential $W(u) = 8u - 16u^2 - \frac{8}{3}u^3 + 8u^4$. This is a double well potential with unequal depth wells; see fig. 1. In this case, equation (5.1) is well-known to possess traveling wave solutions on $x \in \mathbb{R}$, see fig. 2. We choose the initial condition $u(x, 0) = \tanh(4x + 20)$; the exact solution is then $u_*(x, t) = \tanh(4x + 20 - 8t)$. The computational domain is $x \in [-10, 10]$, discretized into a uniform grid of 8193 points. We approximate the solution on $\mathbb{R}$ by using the Dirichlet boundary conditions $u(\pm 10, t) = \pm 1$: The domain size is large enough that the mismatch in boundary conditions does not substantially contribute to the error in the approximate solution over the time interval $t \in [0, 5]$. We use $E_1(u) = \int \frac{1}{2} \|\nabla u\|^2 dx$ and $E_2(u) = \int W(u) dx$. Table 1 tabulates the error in the computed solution at time $T = 5$ for our two new schemes.

Next, we consider the Allen-Cahn equation (5.1) in two space-dimensions, with the potential $W(u) = u^2(1 - u)^2$ that has equal depth wells; see fig. 1. We take the initial condition $u(x, y, 0) = \frac{1}{1 + \exp[-(7.5 - \sqrt{x^2 + y^2})]}$ on the domain $x \in [-10, 10]^2$, and impose periodic boundary conditions. Once again we use $E_1(u) = \int \frac{1}{2} \|\nabla u\|^2 dx$ and $E_2(u) = \int W(u) dx$. As a proxy for the exact solu-

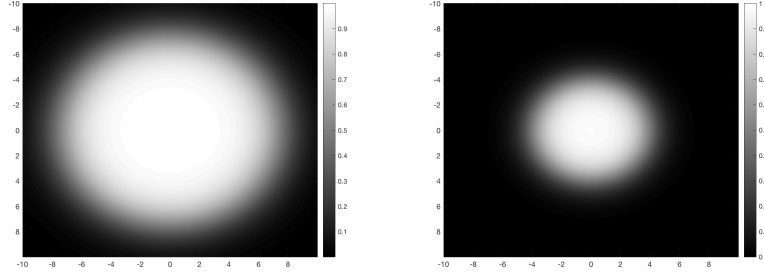


Figure 3: Initial condition and the solution at final time for the 2D Allen-Cahn equation with a potential that has equal depth wells.

tion of the equation with this initial data, we compute a very highly accurate numerical approximation $u_*(x, y, t)$ via the following second order accurate in time, semi-implicit, multi-step scheme [32] on an extremely fine spatial grid and take very small time steps:

$$\frac{3}{2}u_{n+1} - 2u_n + \frac{1}{2}u_{n-1} = k\Delta u_{n+1} - k(2W'(u_n) - W'(u_{n-1})).$$

Table 2 show the errors and convergence rates for the approximate solutions computed by our new multi-stage schemes.

Number of time steps	2^8	2^9	2^{10}	2^{11}	2^{12}
L^2 error (2nd order)	3.62e-05	9.07e-06	2.27e-06	5.68e-07	1.41e-07
Order	-	2.00	2.00	2.00	2.00
L^2 error (3rd order)	2.35e-05	3.18e-06	4.15e-07	5.29e-08	6.24e-09
Order	-	2.88	2.94	2.97	3.08

Table 2: The new second (3.6) and third (3.7) order accurate, conditionally stable schemes (2.1) on the two-dimensional Allen-Cahn equation (5.1) with a potential that has equal depth wells.

For our next example, we consider the Cahn-Hilliard equation

$$u_t = -\Delta(\Delta u - W'(u)) \quad (5.3)$$

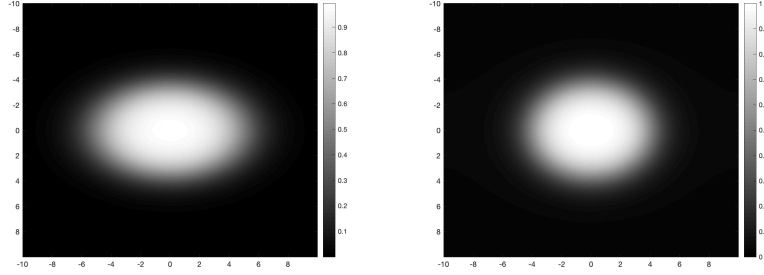


Figure 4: Initial condition and the solution at final time for the 2D Cahn-Hillard equation with a potential that has equal depth wells.

where we take W to be the double well potential $W(u) = u^2(1-u)^2$ with equal depth wells and impose periodic boundary conditions. This flow is also gradient descent for energy (5.2), but with respect to the H^{-1} inner product:

$$\langle u, v \rangle = - \int u \Delta^{-1} v \, dx.$$

Starting from the initial condition $u(x, y, 0) = \frac{1}{1 + \exp[-(5 - \sqrt{x^2 + y^2})]}$, we computed a proxy for the “exact” solution once again using the second order accurate, semi-implicit multi-step scheme from [32]:

$$\frac{3}{2}u_{n+1} - 2u_n + \frac{1}{2}u_{n-1} = -k\Delta[\Delta u_{n+1} - (2W'(u_n) - W'(u_{n-1}))]$$

where the spatial and temporal resolution was taken to be high to ensure the errors are small. Table 3 show the errors and convergence rates for the approximate solutions computed by our new multi-stage schemes.

As a final example we do the following porous medium equation:

$$u_t = \Delta u^{5/3}. \quad (5.4)$$

Under the H^{-1} inner product, (5.4) is gradient flow for the energy

$$E(u) = \frac{3}{8} \int u^{8/3} dx.$$

Our initial data is

$$u(x, 0) = \frac{3}{2\sqrt{2\pi}} \exp\left(-\frac{9x^2}{8}\right) \quad (5.5)$$

Number of time steps	2^7	2^8	2^9	2^{10}	2^{11}
L^2 error (2nd order)	6.20e-04	1.92e-04	5.59e-05	1.55e-05	4.09e-06
Order	-	1.69	1.78	1.85	1.92
L^2 error (3rd order)	6.45e-06	1.35e-06	2.51e-07	4.15e-08	7.20e-09
Order	-	2.25	2.43	2.60	2.53

Table 3: The new second (3.6) and third (3.7) order accurate, conditionally stable schemes (2.1) on the two-dimensional Cahn-Hilliard equation (5.3) with a potential that has equal depth wells.

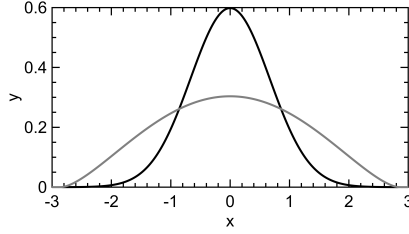


Figure 5: The initial condition (black) and the solution at final time (gray) in the numerical convergence study on the PME

in $x \in [-3, 3]$ with derivative zero Neumann boundary conditions. We let $E_1(u) = \frac{3}{8} \int u^2 dx$ and $E_2(u) = E(u) - \frac{3}{8} \int u^2 dx$. We run the simulation for $T = 1$. See fig. 5 for our initial and final curve. We generate the “true” solution using the L-stable (but not energy stable) TR-BDF2 method with a high spatial and temporal resolution. See table 4 for results.

5.2. Gradient Flow For Solution Dependent Inner Product

Our first example we present in this section is the heat equation, $u_t = \Delta u$, but with a different energy. Under the Wasserstein metric (denoted as W_2), the heat equation is a gradient flow for the negative entropy [33]:

$$E(u) = \int u \log(u) dx. \quad (5.6)$$

Number of time steps	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}
L^2 error (2nd order)	1.91e-06	6.85e-07	2.26e-07	6.90e-08	1.97e-08
Order	-	1.48	1.60	1.71	1.81
L^2 error (3rd order)	2.21e-07	5.69e-08	1.26e-08	2.41e-09	4.13e-10
Order	-	1.96	2.18	2.38	2.54

Table 4: The new second (3.6) and third (3.7) order accurate, conditionally stable schemes (2.1) on the porous medium equation

However the minimization

$$\arg \min_u E(u) + \frac{1}{2k} W_2^2(u, u_n)$$

is a difficult optimization problem. On the other hand, we can approximate the the Wasserstein metric, $W_2(u, v)$, with

$$\langle u - v, \mathcal{L}(u)^{-1}(u - v) \rangle_{L^2} \text{ where } \mathcal{L}(u) = -\nabla \cdot u \nabla \quad (5.7)$$

when u and v are near each other. Indeed

$$-\mathcal{L}(u) \nabla_{L^2} E(u) = \nabla \cdot u \nabla (\log(u) + 1) = \Delta u.$$

Thus, we can alternatively think of the heat equation as minimizing movements on negative entropy with respect to the solution dependent inner product (5.7) and therefore use algorithm 1 and algorithm 2 to evolve the heat equation while decreasing the negative entropy (5.6) at every step.

We use the exact solution $u(x, t) = \cos(\pi x) \exp(-t\pi^2) + 2$ as our test with domain $x \in [0, 1]$ using derivative zero Neumann boundary conditions. Our initial data is $u(x, 0)$ and we run the simulation to final time $T = \frac{1}{10}$. We use $E_1(u) = \frac{1}{2} \int u^2 dx$ and $E_2(u) = \int u \log(u) dx - \frac{1}{2} \int u^2 dx$ in (2.1) so at every step we are solving a linear systems of equation. We run simulation for $T = \frac{1}{10}$. See table 5 for results.

The next example is the porous medium equation in one dimension. The

Number of time steps	2^3	2^4	2^5	2^6	2^7
L^2 error (2nd order)	1.06e-03	3.11e-04	8.58e-05	2.27e-05	5.85e-06
Order	-	1.77	1.86	1.92	1.96
L^2 error (3rd order)	1.00e-05	1.57e-06	2.20e-07	3.04e-08	4.29e-09
Order	-	2.69	2.82	2.87	2.83

Table 5: The new second (algorithm 1) and third (algorithm 2) order accurate, conditionally stable schemes for gradient flows with solution dependent inner product on the heat equation with Wasserstein metric.

Number of time steps	2^4	2^5	2^6	2^7	2^8
L^2 error (2nd order)	2.69e-04	6.10e-05	1.49e-05	3.71e-06	9.25e-07
Order	-	2.14	2.04	2.01	2.00
L^2 error (3rd order)	2.88e-0	3.17e-06	3.72e-07	4.53e-08	5.50e-09
Order	-	3.18	3.09	3.04	3.04

Table 6: The new second and third order accurate, unconditionally stable schemes (see remark 4.1) for gradient flows with solution dependent inner product on the porous medium equation with the linearized Wasserstein metric.

energy is

$$E(u) = \frac{3}{2} \int u^{5/3} dx$$

under the Wasserstein metric. As with the heat equation, we can again replace the Wasserstein metric with (5.7). We will let $E_1(u) = E(u)$ and $E_2(u) = 0$. We use the same test as in the H^{-1} gradient flow porous medium equation (see (5.5) and accompanying explanation). We present the results of the porous medium equation test with movement limiter (5.7) in table 6.

For our final example, we consider the Cahn-Hilliard equation with variable mobility and a forcing term:

$$u_t = -\nabla \cdot \mu(u) \nabla (\epsilon^2 \Delta u - W'(u) - F(x)) \quad (5.8)$$

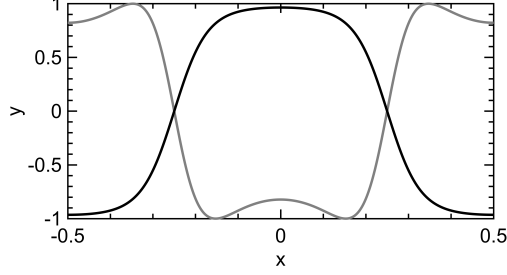


Figure 6: Initial condition (black) and the solution at final time (gray) for the 1D Cahn-Hilliard with variable mobility and forcing term example.

where we take W to be the double well potential $W(u) = (1 - u^2)^2$ with equal depth wells, the forcing term to be $F(x) = \tanh\left(\frac{\cos(2\pi x)}{10\epsilon}\right)$ and the mobility to be $\mu(u) = (1 - \epsilon)(1 - u^2)^2 + \epsilon$ to avoid degeneracy in the PDE. This flow is gradient descent for energy

$$E(u) = \int \frac{\epsilon^2}{2} \|\nabla u\|^2 + W(u) + uF(x) dx,$$

with respect to the solution dependent inner product

$$\langle u - v, \mathcal{L}(u)^{-1}(u - v) \rangle_{L^2} \text{ where } \mathcal{L}(u) = -\nabla \cdot \mu(u) \nabla. \quad (5.9)$$

For our example, we take $\epsilon = \frac{1}{20}$ and start from the initial condition

$$u(x, 0) = \tanh\left(\frac{\cos(2\pi x)}{10\epsilon}\right)$$

on the domain $x \in [-\frac{1}{2}, \frac{1}{2}]$, and impose periodic boundary conditions. We run the PDE until time $T = \frac{1}{8}$. We computed a proxy for the “exact” solution using the following second order BDF/AB scheme:

$$\begin{aligned} 3u_{n+1} + 2k\epsilon^2 \Delta^2 u_{n+1} &= 4u_n - u_{n-1} + 4(k\epsilon^2 \Delta^2 u_n - k\nabla \cdot \mu(u_n) \nabla[\epsilon^2 \Delta u_n - W'(u_n)]) \\ &\quad - 2(k\epsilon^2 \Delta^2 u_{n-1} - k\nabla \cdot \mu(u_{n-1}) \nabla[\epsilon^2 \Delta u_{n-1} - W'(u_{n-1})]) \end{aligned}$$

where the spatial and temporal resolution were taken to be high to ensure the errors are negligible. See fig. 6 for plots of the initial condition and the solution at the final time. Table 7 shows the errors and convergence rates for the approximate solutions computed by our new multi-stage schemes.

Number of time steps	2^7	2^8	2^9	2^{10}	2^{11}
L^2 error (2nd order)	4.10e-04	1.29e-04	3.84e-05	1.09e-05	2.96e-06
Order	-	1.67	1.75	1.82	1.88
L^2 error (3rd order)	1.79e-05	3.68e-06	6.72e-07	1.06e-07	1.42e-08
Order	-	2.28	2.46	2.66	2.90

Table 7: The new second algorithm 1 and third algorithm 2 order accurate, conditionally stable schemes for gradient flows with solution dependent inner product on the one-dimensional Cahn-Hilliard equation with variable mobility and forcing term (5.8).

6. Conclusion

We presented a new class of implicit-explicit additive Runge-Kutta schemes for gradient flows that are high order and conditionally stable. Additionally, we developed new high order stable schemes for gradient flows on solution dependent inner products. Both of these methods allow us to painlessly increase the order of accuracy of existing schemes for gradient flows without sacrificing stability. We provided many numerical examples of gradient flows, including those that have solution dependent inner product, and have shown that the methods achieve their advertised accuracy.

However, in this paper, we have not developed a systematic approach to coming up with conditionally stable methods of a certain order. In fact, there may exist 2nd and 3rd order methods of fewer stages than given here. Additionally, whether these schemes can be used to achieve arbitrarily high (i.e. ≥ 4) order in time is unknown. We leave these questions to future work.

Appendix A. A Second and Third Order Fully Implicit Methods for Gradient Flows with Solution Dependant Inner Product

Here we layout fully implicit, second and third order algorithms for gradient flows with solution dependant inner product,

$$u_t = -\mathcal{L}(u)\nabla E(u).$$

In this case, each substep has form:

$$\left[\sum_{i=0}^{m-1} \gamma_{m,i} \right] U_m + k\mathcal{L}(u_*)\nabla_H E(U_m) = \sum_{i=0}^{m-1} \gamma_{m,i} U_i. \quad (\text{A.1})$$

The satisfaction of consistency to the exact solution, (4.2), is similar to its semi-implicit counterpart in section 4. Theorem 2.1 in [1] (which is the special case of this paper's theorem 2.1 when $E_2 = 0$) ensures that the coefficients in this section form multistage methods that are energy stable. The exact values for the coefficients for the multistage methods can be found at <https://github.com/AZaitzeff/SIgradflow>.

Appendix A.1. Second order example

The following second order method is unconditionally energy stable. Fix a time step size $k > 0$. Set $u_n = u_0$. To obtain u_{n+1} from u_n , carry out the following steps:

1. Find u_* by solving

$$u_* + \frac{1}{2}k\mathcal{L}_n\nabla E(u_*) = u_n$$

2. Find u_{n+1} using (A.1) with coefficients

$$\gamma \approx \begin{pmatrix} 5.0 & 0 & 0 \\ -2.0 & 6.0 & 0 \\ -2.0 & 0.22 & 6.29 \end{pmatrix}. \quad (\text{A.2})$$

and u_* in $\mathcal{L}(u_*)$ given by step 1 of this algorithm.

Appendix A.2. Third order example

This third order algorithm is energy stable as long as

$$\mathcal{L}(u) - \frac{1}{72}k^2 D^2 \mathcal{L}(u)(w, w)$$

is positive definite for all u and w .

Fix a time step size $k > 0$. Set $u_n = u_0$. For convenience, we will denote $D^2 \mathcal{L}(u_*)(\mathcal{L}(u_*) \nabla E(u_*), \mathcal{L}(u_*) \nabla E(u_*))$ as $D^2 \mathcal{L}(u_*)$ and $MS(\tilde{k}, \mathcal{L}(u_*), \tilde{u}, \gamma)$ as U_M in the multistage algorithm

$$\left[\sum_{i=0}^{m-1} \gamma_{m,i} \right] U_m + \tilde{k} \mathcal{L}(u_*) \nabla_H E(U_m) = \sum_{i=0}^{m-1} \gamma_{m,i} U_i.$$

with time step $\tilde{k}$, operator $\mathcal{L}(u_*)$, $U_0 = \tilde{u}$ and coefficients γ . To obtain u_{n+1} from u_n , carry out the following steps:

1. Set $\gamma = (1)$ and $u_{*1} = MS\left(\frac{1}{6}k, \mathcal{L}(u_n), u_n, \gamma\right)$.
2. Set

$$\gamma \approx \begin{pmatrix} 11.17 & 0 & 0 & 0 & 0 & 0 \\ -7.5 & 19.43 & 0 & 0 & 0 & 0 \\ -1.05 & -4.75 & 13.98 & 0 & 0 & 0 \\ 1.8 & 0.05 & -7.83 & 13.8 & 0 & 0 \\ 6.2 & -7.17 & -1.33 & 1.63 & 11.52 & 0 \\ -2.83 & 4.69 & 2.46 & -11.55 & 6.68 & 11.95 \end{pmatrix} \quad (\text{A.3})$$

and

$$\bar{u} = MS\left(\frac{1}{2}k, \mathcal{L}(u_{*1}) - \frac{1}{72}D^2 \mathcal{L}(u_{*1}), u_n, \gamma\right).$$

3. Set $\gamma = (1)$ and $u_{*2,1} = MS\left(\frac{2}{5}k, \mathcal{L}(u_n), u_n, \gamma\right)$.
4. Set

$$\gamma \approx \begin{pmatrix} 6.17 & 0 & 0 & 0 \\ -0.5 & 6 & 0 & 0 \\ -3 & 2 & 7 & 0 \\ -3.1 & 0 & 2.23 & 7.40 \end{pmatrix} \quad (\text{A.4})$$

and

$$u_{*2,2} = MS\left(\frac{5}{6}k, \mathcal{L}(u_{*2,1}), u_n, \gamma\right).$$

5. Set γ to (A.3) and then

$$u_{n+1} = MS\left(\frac{1}{2}k, \mathcal{L}(u_{*2,2}) - \frac{1}{72}D^2\mathcal{L}(u_{*2,2}), \bar{u}, \gamma\right).$$

Acknowledgments

Funding: Alexander Zaitzeff gratefully acknowledges support from the NSF grant DMS-1719727. Selim Esedoglu was supported by NSF grants DMS-1719727 and DMS-2012015. Krishnna Garikipati acknowledges NSF grant DMREF-1729166.

References

- [1] A. Zaitzeff, S. Esedoglu, K. Garikipati, Variational extrapolation of implicit schemes for general gradient flows, *SIAM Journal on Numerical Analysis* 58 (2020) 2799–2817.
- [2] A. Zaitzeff, S. Esedoglu, K. Garikipati, Second order threshold dynamics schemes for two phase motion by mean curvature, *Journal of Computational Physics* 410 (2020) 109404.
- [3] J. Shin, H. G. Lee, J.-Y. Lee, Unconditionally stable methods for gradient flow using convex splitting Runge–Kutta scheme, *J. Comput. Phys.* 347 (2017) 367–381.
- [4] J. Shin, H. G. Lee, J.-Y. Lee, Convex splitting Runge-Kutta methods for phase-field models, *Computers & Mathematics with Applications* 73 (2017) 2388–2403.
- [5] D. J. Eyre, Unconditionally gradient stable time marching the Cahn–Hilliard equation, *MRS Online Proceedings Library Archive* 529 (1998) 39–46.

- [6] C. Xu, T. Tang, Stability analysis of large time-stepping methods for epitaxial growth models, *SIAM Journal on Numerical Analysis* 44 (2006) 1759–1779.
- [7] J. Shen, X. Yang, Numerical approximations of Allen-Cahn and Cahn-Hilliard equations, *Discrete and Continuous Dynamical Systems* 28 (2010) 1669–1691.
- [8] K. Glasner, S. Orizaga, Improving the accuracy of convexity splitting methods for gradient flow equations, *Journal of Computational Physics* 315 (2016) 52–64.
- [9] Z. Hu, S. M. Wise, C. Wang, J. S. Lowengrub, Stable and efficient finite-difference nonlinear-multigrid schemes for the phase field crystal equation, *Journal of Computational Physics* 228 (2009) 5323–5339.
- [10] S. M. Wise, C. Wang, J. S. Lowengrub, An energy-stable and convergent finite-difference scheme for the phase field crystal equation, *SIAM Journal on Numerical Analysis* 47 (2009) 2269–2288.
- [11] Q. Du, R. Nikolaides, Numerical analysis of a continuum model of a phase transition, *SIAM Journal on Numerical Analysis* 28 (1991) 1301–1322.
- [12] L. Ju, X. Li, Z. Qiao, Energy stability and convergence of exponential time differencing schemes for the epitaxial growth model without slope selection, *Mathematics of Computation* 87 (2018) 1859–1885.
- [13] X. Wang, L. Ju, Q. Du, Efficient and stable exponential time differencing Runge-Kutta methods for phase field elastic bending energy models, *Journal of Computational Physics* 316 (2016) 21–38.
- [14] C. M. Elliott, D. A. French, F. A. Milner, A second order splitting method for the Cahn-Hilliard equation, *Numerische Mathematik* 54 (1989) 575–590.

- [15] J. Shen, C. Wang, X. Wang, S. M. Wise, Second order convex splitting schemes for gradient flows with Ehrlich-Schwoebel type energy: Application to thin film epitaxy, *SIAM Journal on Numerical Analysis* 50 (2012) 105–125.
- [16] Z. Guan, J. S. Lowengrub, C. Wang, S. M. Wise, Second order convex splitting schemes for periodic nonlocal Cahn-Hilliard and Allen-Cahn equations, *Journal of Computational Physics* 277 (2014) 48–71.
- [17] Z. Guan, C. Wang, S. M. Wise, A convergent convex splitting scheme for the periodic nonlocal Cahn-Hilliard equation, *Numerische Mathematik* 128 (2014) 377–406.
- [18] J. Guo, C. Wang, S. M. Wise, X. Yue, An H^2 convergence of a second-order convex-splitting scheme for the modified phase field crystal equation, *Communications in Mathematical Sciences* 14 (2016) 489–515.
- [19] A. Baskaran, J. S. Lowengrub, C. Wang, S. Wise, Convergence analysis of a second order convex splitting scheme for the modified phase field crystal equation, *SIAM Journal on Numerical Analysis* 51 (2013) 2851–2873.
- [20] A. Aristotelous, O. Karakasian, S. M. Wise, Adaptive, second-order in time primitive-variable discontinuous Galerkin schemes for Cahn-Hilliard equation with a mass source term, *IMA Journal of Numerical Analysis* 35 (2015) 1167–1198.
- [21] K. Cheng, Z. Qiao, C. Wang, A third order exponential time differencing numerical schemes for no-slope election thin film model with energy stability, *Journal of Scientific Computing* 81 (2019) 154–185.
- [22] W. Chen, W. Li, C. Wang, S. Wang, X. Wang, Energy stable high-order linear ETD multi-step methods for gradient flows: application to thin film epitaxy, *Research in the Mathematical Sciences* 7 (2020).
- [23] Y. Hao, Q. Huang, C. Wang, A third order BDF energy stable linear scheme for the no-slope-selection thin film model, 2020. arXiv:2011.01525.

- [24] W. Chen, C. Wang, X. Wang, S. M. Wise, Positivity-preserving, energy stable numerical schemes for the Cahn-Hilliard equation with logarithmic potential, *J. Comput. Phys.* X 3 (2019) 100031.
- [25] D. Han, X. Wang, A second order in time, uniquely solvable, unconditionally stable numerical scheme for Cahn–Hilliard–Navier–Stokes equation, *J. Comput. Phys.* 290 (2015) 139–156.
- [26] F. Del Teso, J. Endal, E. R. Jakobsen, Robust numerical methods for nonlocal (and local) equations of porous medium type. part ii: Schemes and experiments, *SIAM J. Numer. Anal.* 56 (2018) 3611–3647.
- [27] C. Duan, C. Liu, C. Wang, X. Yue, Numerical methods for porous medium equation by an energetic variational approach, *J. Comput. Phys.* 385 (2019) 13–32.
- [28] M. Westdickenberg, J. Wilkening, Variational particle schemes for the porous medium equation and for the system of isentropic euler equations, *ESAIM Math. Model. Numer. Anal.* 44 (2010) 133–166.
- [29] C. A. Kennedy, M. H. Carpenter, Additive Runge–Kutta schemes for convection–diffusion–reaction equations, *Appl. Numer. Math.* 44 (2003) 139–181.
- [30] L. Pareschi, G. Russo, Implicit–explicit Runge–Kutta schemes and applications to hyperbolic systems with relaxation, *J. Sci. Comput.* 25 (2005) 129–155.
- [31] E. Zharovsky, A. Sandu, H. Zhang, A class of implicit-explicit two-step Runge–Kutta methods, *SIAM J. Math. Anal.* 53 (2015) 321–341.
- [32] L. Q. Chen, J. Shen, Applications of semi-implicit Fourier-spectral method to phase field equations, *Comput. Phys. Commun.* 108 (1998) 147–158.
- [33] R. Jordan, D. Kinderlehrer, F. Otto, The variational formulation of the Fokker–Planck equation, *SIAM J. Math. Anal.* 29 (1998) 1–17.