Commun. Comput. Phys.
doi: 10.4208/cicp.290414.231214a

Vol. **18**, No. 1, pp. 203-229
July 2015

A Fast Semi-Implicit Level Set Method for Curvature Dependent Flows with an Application to Limit Cycles Extraction in Dynamical Systems

Guoqiao You[1] and Shingyu Leung[2,*]

[1] *The School of Science, Nanjing Audit University, Nanjing, Jiangsu Province, China.*
[2] *Department of Mathematics, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong.*

Communicated by Bo Li

Received 29 April 2014; Accepted (in revised version) 23 December 2014

Abstract. We propose a new semi-implicit level set approach to a class of curvature dependent flows. The method generalizes a recent algorithm proposed for the motion by mean curvature where the interface is updated by solving the Rudin-Osher-Fatemi (ROF) model for image regularization. Our proposal is general enough so that one can easily extend and apply the method to other curvature dependent motions. Since the derivation is based on a semi-implicit time discretization, this suggests that the numerical scheme is stable even using a time-step significantly larger than that of the corresponding explicit method. As an interesting application of the numerical approach, we propose a new variational approach for extracting limit cycles in dynamical systems. The resulting algorithm can automatically detect multiple limit cycles staying inside the initial guess with no condition imposed on the number nor the location of the limit cycles. Further, we also propose in this work an Eulerian approach based on the level set method to test if the limit cycles are stable or unstable.

AMS subject classifications: 37A25, 37M25, 76M27

Key words: Numerical methods for PDEs, level set method, dynamical systems, flow visualization.

1 Introduction

Curvature dependent flows are interesting not only mathematically but also computationally. Numerically, the motion of a parametrized curve can be determined by solving the corresponding ordinary differential equation (ODE) for each discretized point

[*] Corresponding author. *Email addresses:* youguoqiao@sina.com (G. You), masyleung@ust.hk (S. Leung)

on the curve. However, as the curve evolves, such explicit representation might require re-meshing to obtain a better interface sampling. And worst, it might also need careful numerical surgery if there is a topological change when the curve splits into multiple disjoint components. Another class of numerical algorithms is implicit methods based on the level set approach, including the approach in [44] by regularizing the curvature term using the Laplacian of the level set function, a modified MBO approach [12] which generates appropriate motion by diffusion, and the variational approach in [7,8] by applying the ROF functional [40] from the image processing community.

In this paper, we propose a new semi-implicit scheme for computing a class of curvature dependent evolutions of a codimension-one surface in the level set formulation. We consider the evolution of the level set equation

$$\phi_t = |\nabla\phi| v_n(\kappa) = |\nabla\phi| v_n \left[\nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|} \right) \right], \tag{1.1}$$

where the normal velocity $v_n(\kappa)$ defined on the interface satisfies $v_n(\kappa)\kappa \geq 0$. Mathematically, this constraint on the normal velocity gives a stability condition in the curve evolution. To see this, one can show that (for example in [41]) if $\alpha(s)$ denotes the arclength of a curve $\gamma(s;t) = \{(x(s;t),y(s;t)) : t \geq 0\}$ parametrized by the parameter s at a given time t, then we have $d\alpha = g(s;t)ds$ where $g(s;t) = \sqrt{x_s^2 + y_s^2}$ and

$$g_t(s;t) = -g(s;t) v_n(\kappa)\kappa.$$

Therefore, the condition $v_n(\kappa)\kappa \geq 0$ actually imposes a condition that the curve should collapse under its evolution in time.

Our approach is developed based on a regularization technique. However, unlike the regularization by a standard Laplacian as in the approach in [44], we propose to regularize the geometrical flow by adding and subtracting a *curvature* term. Applying the algorithm to the motion by mean curvature, we will show that our scheme reduces to the variational functional in [7, 8]. From this point of view, our method can be regarded as a generalization of [7, 8]. However, unlike these approaches, our interpretation allows us to easily extend the algorithm to deal with a much wider class of curvature dependent flows. Moreover, since our approach is developed based on a semi-implicit time-discretization, the resulting numerical method has a relatively large time-step size. Even though we do not have any theoretical estimate on the required stability condition since the regularization is in fact a nonlinear one, we find that the numerical solution gives a stable evolution of the interface with a marching step significantly larger than the one by a corresponding explicit scheme. This interesting property has not yet been reported in the work of [7, 8] which is solely based on the property of the ROF functional.

As an interesting and important application of the proposed algorithm, we develop and apply a variational method for extracting invariant manifolds in dynamical systems. One kind of important invariant manifolds is the Lagrangian coherent structures (LCS). The main idea in determining the LCS is to partition the space-time domain into different regions according to a Lagrangian quantity advected along with passive tracers. One of many possible Lagrangian quantities is the finite time Lyapunov exponent (FTLE) [18, 19, 42]. Numerically, the first step to compute the FTLE is to move particles in the flow for a period of time and obtain the flow map which takes the initial particle location to its arrival location. Mathematically, the motion of these particles in the extended phase space satisfies a given dynamical system

$$\dot{\mathbf{x}}(t)=\mathbf{u}(\mathbf{x}(t),t) \tag{1.2}$$

with a given Lipschitz velocity field $\mathbf{u}:\mathbb{R}^d\times\mathbb{R}\rightarrow\mathbb{R}^d$ and an initial condition $\mathbf{x}(t_0)=\mathbf{x}_0$. We define the flow map $\Phi_{t_0}^T:\mathbb{R}^d\rightarrow\mathbb{R}^d$ to be the mapping which takes the point $\mathbf{x}_0$ to the particle location at the final time $t=t_0+T$, i.e. $\Phi_{t_0}^T(\mathbf{x}_0)=\mathbf{x}(t_0+T)$ with $\mathbf{x}(t)$ satisfying (1.2). Then the FTLE is computed from the Jacobian of the resulting flow map. Based on the level set method [38] and the backward phase flow method [29], we have developed in [28] numerical methods for moderate to long time FTLE computations which lead to a more efficient computation of FTLE. Developed based on those algorithms in [27, 28], we have recently proposed in [47] an efficient Eulerian numerical approach to extract invariant sets in a continuous dynamical system in the extended phase space (the $\mathbf{x}-t$ space). We have extended the idea of ergodic partition and have proposed a concept called *coherent ergodic partition* for visualizing ergodic components in a continuous flow. Another Eulerian method to study dynamical systems can be found in [48].

In this paper, we are going to extend these Eulerian approaches to extract another kind of invariant manifolds in the flow. In practice, one of the most widely studied invariant manifolds is still the *limit cycle* in a dynamical system, likely because of the second part of the Hilbert's XVI-th problem [21]. The problem asked for both the number and the location of limit cycles in a two dimensional dynamical system with polynomial vector fields of degree-n. Recent theoretical development and review on the problem can be found in [11, 22], and thereafter. Even though there is still no definitely theoretical answer to the original problem, several numerical approaches have been proposed to approximate the location of such invariant set. For example, [32] has proposed a variational approach for limit cycle extraction by minimizing the functional

$$\min_{\gamma(s)}\int_s\|\gamma'(s)\times\mathbf{u}(\gamma(s))\|^2ds \tag{1.3}$$

with respect to the curve $\gamma(s)$, which aims to determine a curve $\gamma(s)$ such that its tangent directions ally with the velocity field. Numerically, the parametrized curve $\gamma(s)$ is first discretized and then the functional (1.3) is minimized using the gradient descent with an initial guess of the limit cycle. Such approach is effective when there is only one single limit cycle enclosed by the initial guess. When there are several disjointed limit cycles in the domain of interest, surgery has to be done numerically to the parametrization γ_i in order to take care of the interface splitting in the gradient descent. More discussions can be found in some similar applications in image segmentation using the snake model [24] or the geodesic active contour model [6].

In this work, we follow a similar approach but propose a new variational model for extracting the limit cycles based on the level set method. The idea is to introduce a level set function whose zero level set implicitly represents the limit cycle, while the evolution of the zero level set is done implicitly by solving a corresponding level set equation. To develop a computationally efficient method for the resulting variational formulation based on the level set method and the gradient descent, we incorporate the semi-implicit solver by treating the curvature term implicitly. The resulting algorithm can automatically detect multiple limit cycles staying inside the initial guess. One does not impose any condition on the number nor the location of the limit cycles. Further, we also propose in this work an Eulerian approach to test the stability of the limit cycles based on the level set method.

The rest of the paper is organized as follows. In Section 2, we will first summarize various main approaches to the motion by mean curvature. We will compare these approaches and will conclude that they are all related through variational formulations. In Section 3.1, we will give our proposed semi-implicit scheme for solving the motion by mean curvature. In Section 3.2, we will generalize the numerical approach in Section 3.1 to any other curvature dependent flow. As an interesting application to the algorithm, we propose a new variational formulation to extract limit cycles in a dynamical system in Section 4. Extensive numerical experiments will finally be given in Section 5 to demonstrate the stability and effectiveness of the proposed method.

2 Previous approaches to the motion by mean curvature

We first concentrate on the level set equation for the motion by mean curvature given by

$$\phi_t = |\nabla\phi|\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right) = |\nabla\phi|\kappa, \tag{2.1}$$

where $\kappa=\nabla\cdot(\nabla\phi/|\nabla\phi|)$ is the mean curvature [37]. This nonlinear equation can be easily solved by the simple explicit scheme given by

$$\frac{\phi^{k+1}-\phi^k}{\Delta t} = |\nabla\phi^k|\nabla\cdot\left(\frac{\nabla\phi^k}{|\nabla\phi^k|}\right), \tag{2.2}$$

where all derivatives are approximated by central difference. Since the curvature term involves the second derivative of the level set function, the time step constraint for this explicit scheme is of order $\Delta t = \mathcal{O}(\Delta x^2)$. This results in a computationally inefficient numerical method.

If ϕ^k is reinitialized at each intermediate step such that $|\nabla\phi^k|=1$, (2.2) can be reduced to the heat equation

$$\frac{\phi^{k+1}-\phi^k}{\Delta t} = \Delta\phi^k.$$

This approach has been discussed in [37] and a finite element method has also been implemented in [25].

To relax the stability constrain in the explicit scheme, one might use the following simple semi-implicit scheme

$$\frac{\phi^{k+1}-\phi^k}{\Delta t}=|\nabla\phi^k|\nabla\cdot\left(\frac{\nabla\phi^{k+1}}{|\nabla\phi^k|}\right). \tag{2.3}$$

However, to update ϕ^{k+1} from ϕ^k for each k, one has to reconstruct a sparse symmetric positive definite matrix and invert the resulting large system of linear equations. This could be time-consuming in practice.

The discussion in the rest of this section is definitely not meant to be a complete one but we would like to concentrate the discussion only on methods related to the one we are going to propose. For instance, we are not going to discuss the variational approach in [1], the original MBO method based on the diffusion of a Heaviside function [5,33], a split Bregman approach for crystalline mean curvature flow [35], and etc.

2.1 Regularization by the Laplacian

To relax the time step restriction in the explicit scheme for the motion by the mean curvature, [44] has proposed to regularize the flow using the Laplace operator by adding and subtracting the Laplacian of the level set function to the evolution equation. In particular, the paper first rewrites the evolution (2.1) as

$$\phi_t=\Delta\phi-(\Delta\phi-|\nabla\phi|\kappa)=\Delta\phi-\mathbf{n}\cdot\nabla(|\nabla\phi|)=\Delta\phi-\mathcal{N}(\phi), \tag{2.4}$$

where $\mathcal{N}(\phi)=\mathbf{n}\cdot\nabla(|\nabla\phi|)$ is a nonlinear term which measures the difference between the curvature motion term and the linear Laplacian. Following usual explicit-implicit methods or usual semi-implicit methods, the paper proposes to treat the linear part of (2.4) implicitly and the nonlinear part explicitly. This gives

$$\frac{\phi^{k+1}-\phi^k}{\Delta t}=\Delta\phi^{k+1}-\mathcal{N}(\phi^k)$$

with $\Delta t\gg\mathcal{O}(\Delta x^2)$ and therefore ϕ^{k+1} can be obtained by

$$\phi^{k+1}=(I-\Delta t\Delta)^{-1}[\phi^k-\Delta t\mathcal{N}(\phi^k)]. \tag{2.5}$$

Since the operator $(I-\Delta t\Delta)$ is linear, one can invert it easily using FFT. The importance of this approach is that it *regularizes* the nonlinear evolution by a linear operator which can be efficiently solved.

A similar development has recently been proposed in the phase-field community. For example, to stabilize the evolution of the Cahn-Hilliard equation, [20] has also proposed to add-and-subtract a term $\mathcal{O}(\Delta\phi)$. Then the paper treats one of them implicitly while the other one explicitly. Stability and convergence of the approach have also been discussed in that paper.

2.2 Diffusion generated motion

A diffusion generated scheme is given in [12] using signed distance functions. The idea is to minimize the functional

$$E_{DGM}(\phi)=\int|\nabla\phi|^2+\frac{1}{\Delta t}|\phi-d(\phi^k)|^2, \tag{2.6}$$

where $d(\phi^k)$ represents the signed distance function obtained by reinitializing ϕ^k, i.e. $|\nabla d(\phi^k)|=1$ such that ϕ^k and $d(\phi^k)$ share the same zero level set. Numerically, various ways can be done to determine $d(\phi^k)$ from ϕ^k. A simple way is to solve the following partial differential equation (PDE)

$$\frac{\partial\tilde{\phi}}{\partial\tau}+\text{sgn}(\phi^k)(|\nabla\tilde{\phi}|-1)=0$$

in the artificial time direction τ with $\tilde{\phi}(x;\tau=0)=\phi^k$. Higher order WENO-type methods have been proposed to accurately solve this reinitialization equation [23].

2.3 Rudin-Osher-Fatemi (ROF) model

Relating to (2.6) is the approach in [7,8] given by

$$E_C(\phi)=\int|\nabla\phi|+\frac{1}{2\Delta t}|\phi-d(\phi^k)|^2, \tag{2.7}$$

which replaces the Tikhonov regularization by the total variation (TV) norm of ϕ. The functional relates to the so-called Rudin-Osher-Fatemi (ROF) model for image restoration [40]. Given an observed noisy image f, one regularizes it by looking for a bounded variation (BV) function u such that it solves

$$\min_u\int|\nabla u| \tag{2.8}$$

with $\|u-f\|^2=\sigma^2$ representing the variance of the Gaussian white noise. The semi-norm (2.8) is called the TV norm. In practice, one introduces a Lagrange multiplier λ and obtains the following ROF functional

$$E_{ROF}(u)=\int|\nabla u|+\frac{\lambda}{2}|u-f|^2. \tag{2.9}$$

There are extensive successful results in the image processing community based on this TV-regularization. Tremendous research has been stimulated since the paper [40] which has proposed to use such norm for image regularization, see for example [4,10,37] and thereafter.

[7,8] has proposed to minimize the functional (2.7) at any intermediate time $t=t^k$ such that the Lagrange multiplier λ in the ROF model is chosen to be $1/\Delta t$ and the observed image f is the signed distance function of ϕ^k. Therefore, the method alternatively minimizes (2.7) and reinitializes the minimizer ϕ^{k+1} to obtain $d(\phi^{k+1})$. A convergence proof of this numerical scheme is given in [7].

2.4 Comparison of various approaches

We note that all these approaches are related to variational functionals with similar forms. It is obvious for the last two approaches (2.6) and (2.7) and we do not further comment on this here. For the approach proposed in [44] based on the regularization using a Laplacian, the evolution (2.5) can actually be rewritten as the following minimization problem

$$E_S(\phi)=\int|\nabla\phi|^2+\frac{1}{\Delta t}|\phi-f(\phi^k)|^2 \tag{2.10}$$

with $f(\phi)=\phi-\Delta t\mathcal{N}(\phi)$. Given ϕ^k, one minimizes this functional to get ϕ^{k+1}. We can see that the minimization problem (2.10) is the same with (2.6) except the input function f.

Moreover, there is another relationship between the functionals (2.6) and (2.10). We have observed that in the case if [44] reinitializes any intermediate solution ϕ^k such that $|\nabla\phi^k|=1$, we have

$$\mathcal{N}(\phi^k)=\mathbf{n}\cdot\nabla(|\nabla\phi^k|)=\mathbf{n}\cdot\nabla(1)=0$$

and so $f(\phi^k)=\phi^k-\Delta t\mathcal{N}(\phi^k)=\phi^k=d(\phi^k)$. This implies that the scheme (2.10) can be interpreted as a generalization to (2.6).

To end this section, we consider again the algorithm for (2.10). To update ϕ^{k+1} from ϕ^k, we first solve the nonlinear evolution equation

$$\phi_\tau+\mathcal{N}(\phi)=0 \tag{2.11}$$

using an explicit method for **one** single time step using $\Delta\tau\gg\mathcal{O}(\Delta x^2)$. This gives $f(\phi^k)$. The next step is to plug in this intermediate solution to the functional (2.10). The minimizer to the energy then gives ϕ^{k+1}. Note that the magnitude Δt in obtaining $f(\phi^k)$ clearly violates the corresponding stability condition for the evolution equation (2.11). However, such unstable evolution is then regularized by the Tikhonov functional to give a stable approximation to the motion by mean curvature (2.1).

3 A fast semi-implicit method

3.1 Motion by mean curvature

In this section, we introduce a semi-implicit scheme for solving the motion by mean curvature in the level set method. At each step, we reformulate the iteration as a convex optimization problem which can be solved efficiently using any recent fast algorithms. These include the second order cone programming method [14], the fixed-point continuation approach [17], methods based on the Bregman iterations [15,36], other primal-dual approach or the Arrow-Hurwicz method [3,9,26,50], the primal-dual approach [9,13,39], and etc.

Let $\Delta t\gg\mathcal{O}(\Delta x^2)$ be a given time-step at time $t=t^k$. To approximate the solution at $t=t^{k+1}$, we propose to regularize the flow by adding and subtracting a *curvature* term, i.e. we consider

$$\phi_t = \beta\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \beta\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right) + |\nabla\phi|\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right),$$

for some constant $\beta > 0$ which will be further discussed later this section. Numerically, we treat the first curvature term implicitly and the second one explicitly. This gives the following regularized discretization

$$\frac{\phi^{k+1}-\phi^k}{\Delta t} = \beta\nabla\cdot\left(\frac{\nabla\phi^{k+1}}{|\nabla\phi^{k+1}|}\right) - \beta\nabla\cdot\left(\frac{\nabla\phi^k}{|\nabla\phi^k|}\right) + |\nabla\phi^k|\nabla\cdot\left(\frac{\nabla\phi^k}{|\nabla\phi^k|}\right), \tag{3.1}$$

where the term $|\nabla\phi|$ in the denominator is numerically regularized by $\sqrt{\phi_x^2+\phi_y^2+\epsilon^2}$.

This discretization can be reformulated as an energy minimization problem. For instance, to determine ϕ^{k+1} in (3.1), we minimize

$$\int |\nabla\phi| + \frac{\lambda}{2}|\phi - f(\phi^k)|^2 \tag{3.2}$$

for some constant $\beta > 0$ with $\lambda = (\beta\Delta t)^{-1}$ and

$$f(\phi) = \phi + \Delta t(|\nabla\phi| - \beta)\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right). \tag{3.3}$$

We observe that the function $f(\phi^k)$ defined in (3.3) can be interpreted as an intermediate solution to

$$\phi_\tau = (|\nabla\phi| - \beta)\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right) \tag{3.4}$$

obtained by marching **one** explicit forward Euler step with the time step $\Delta\tau \gg \mathcal{O}(\Delta x^2)$. Numerically, even if (3.4) is well-posed (which might not be true if $\beta \gg 1$), one does not expect the corresponding evolution

$$\phi^k = f(\phi^{k-1}) = f^2(\phi^{k-2}) = \cdots = f^k(\phi^0)$$

would give a stable approximation to (3.4) (**not** (2.1)) at time $t = t^k$. However, if the solution is then regularized by the ROF functional with a suitable β, the numerical algorithm would result in a good approximation to the motion by mean curvature.

Now, we consider a way to determine the positive quantity β. The main idea of adding and subtracting the term $\beta\kappa$ in (3.1) is to extract off a part from the nonlinear term $|\nabla\phi^k|\kappa$ which will be treated implicitly. Numerically we propose to determine β by minimizing the right-hand side of equation (3.4) over Ω in the L_2 sense, i.e. β is chosen to solve the following optimization problem

$$\min_{\beta>0} \int_\Omega \left[(\beta - |\nabla\phi^k|)\kappa\right]^2$$

which gives

$$\beta = \frac{\int_\Omega |\nabla\phi^k| \kappa^2 d\mathbf{x}}{\int_\Omega \kappa^2 d\mathbf{x}} \geq 0.$$

To the best of our knowledge, this algorithm gives a new semi-implicit scheme for the motion by mean curvature. However, the resulting algorithm might indeed relate to some other developed methods. To see this, if one further reinitializes all intermediate solutions to make the level set function ϕ^k a signed distance function, i.e. $|\nabla\phi^k| = 1$, we have $\beta = 1$ and $f(\phi^k) = d(\phi^k)$. Therefore, (3.2) reduces to the same algorithm proposed in [7,8].

Indeed there is an advantage of reinitializing all intermediate solutions in our derivation. Since all one cares is the zero level set, one can actually concentrate all computational power within a small neighborhood of $\{x : \phi^k(x) = 0\}$. Instead of minimizing (2.7) over the whole computational domain Ω, one can first determine a computational tube $\Gamma^k = \{x : |\phi(x;t^k)| < \gamma\}$ for each time step t^k and then minimize the functional (2.7) over only Γ^k rather than Ω. This leads to the following algorithm.

Algorithm 3.1: Local Level Set Semi-Implicit Method for the Motion by Mean Curvature

```
1. Initialization. Given $\gamma = \mathcal{O}(\Delta x)$, $\Delta t \gg \mathcal{O}(\Delta x^2)$ and a signed distance function $\phi(x;t^0)$ with the
   initial interface given by $\{x : \phi(x;t^0) = 0\}$.
2. Iteration. For $k = 0,1,\cdots,N$
   (a) Determine the computation tube $\Gamma^k = \{x : |\phi(x;t^k)| < \gamma\}$.
   (b) Solve the ROF functional (2.9) with $\lambda = (\Delta t)^{-1}$ and $f = \phi(x;t^k)$ in $\Gamma^k$.
   (c) Reinitialize the solution for a few steps to get $\phi(x;t^{k+1})$.
```

Although this local level set method can speed up the whole computational complexity by a factor of N where N is the number of mesh points in one dimension, the choice of the radius of the tube γ could depend on Δt and also the number of steps in the reinitialization process. As a result, we leave it as a future work and we are going to use the following full level set version in this paper instead.

Algorithm 3.2: Full Level Set Semi-Implicit Method for the motion by mean curvature

```
1. Initialization. Initialize $\Delta t \gg \mathcal{O}(\Delta x^2)$ and a signed distance function $\phi(x;t^0)$ with the initial
   interface given by $\{x : \phi(x;t^0) = 0\}$.
2. Iteration. For $k = 0,1,\cdots,N$
   (a) Minimize the ROF functional (3.2) with $\lambda = (\beta\Delta t)^{-1}$ and $f = f(\phi^k)$ given by (3.3).
   (b) Reinitialize the solution for a few steps to get $\phi(x;t^{k+1})$.
```

Even though the approach in this section is only a small generalization of the one proposed in [7, 8], we have introduced a different derivation from the point of view of a semi-implicit scheme for the equation. As we will see later in the next section, such derivation provides a simple and natural way to extend the regularization technique to a wide class of curvature dependent flows.

3.2 Motions depending on curvature

In this section, we are going to extend Algorithm 3.2 to other curvature dependent motions. In the level set formulation, the evolution of the interface under the flow $v_n = v_n(\kappa)$ (assuming that $v_n \cdot \kappa \geq 0$) can be computed by solving the level set equation (1.1). Generalizing the above approach, we regularize this evolution by adding and subtracting a curvature term with some weight $\beta > 0$. This gives

$$\phi_t = |\nabla\phi| v_n(\kappa) = \beta\kappa - \beta\kappa + |\nabla\phi| v_n(\kappa).$$

Numerically, we treat the first curvature term implicitly and the second one explicitly. Therefore, ϕ^{k+1} can be determined by minimizing the same functional (3.2) with $\lambda = (\beta\Delta t)^{-1}$ and

$$f(\phi) = \phi + \Delta t \left\{ |\nabla\phi| v_n \left[\nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|} \right) \right] - \beta \nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|} \right) \right\}.$$

Once again, the amount of the regularization (i.e. the magnitude of β) is chosen to minimize the L_2 difference in the update formula in $f(\phi)$, i.e. we minimize the following function with respect to β

$$\min_{\beta>0} \int_\Omega [|\nabla\phi| v_n - \beta\kappa]^2.$$

This leads to

$$\beta = \frac{\int_\Omega |\nabla\phi| v_n(\kappa)\kappa d\mathbf{x}}{\int_\Omega \kappa^2 d\mathbf{x}},$$

which is always positive if $v_n(\kappa)\kappa > 0$.

4 Application to limit cycles extraction

As an important and interesting application of the proposed semi-implicit scheme, we are going to develop a fast numerical approach to determine both the number and the location of limit cycles of a dynamical system (in which the velocity field is not necessary to be in the form of a polynomial as described in the original Hilbert's problem) in a finite computational domain.

Definition 4.1. We consider a planar dynamical system given by $\mathbf{x}'(t) = \mathbf{u}(\mathbf{x}(t))$ where $\mathbf{u} : \mathbb{R}^2 \to \mathbb{R}^2$ is a smooth function. A closed trajectory $\mathbf{x}(t)$ satisfying the system is called a **limit cycle** if at least one other trajectory spirals into it either as time approaches positive infinity or as time approaches negative infinity. If all nearby trajectories spiral into this closed curve as time goes to infinity (negative infinity), we call it a **stable** (an **unstable**) limit cycle.

Our idea is developed based on [45,46] for geometric surface processing via normal maps. Assuming that a closed curve is implicitly represented as the zero level set of a function $\phi(\mathbf{x}):\Omega\to\mathbb{R}$, [45,46] have proposed to iteratively regularize the following two steps:

1. Obtain the normal vector field $\mathbf{n}^{k+1}$ with the level set function ϕ^k fixed by minimizing
$$E_{\mathbf{n}}(\mathbf{n})=\int_\Omega G(|(\nabla\mathbf{n})[\mathbf{I}-\mathbf{P}(\phi^k)]|^2),$$
where $\mathbf{I}$ is the identity matrix, $\mathbf{P}$ is the projection matrix, and $G'(x)$ is the edge stopping function.
2. Determine the level set function ϕ^{k+1} with the normal vector field $\mathbf{n}^{k+1}$ fixed by minimizing
$$E_\phi(\phi)=\int_\Omega|\nabla\phi|\left(1-\frac{\nabla\phi}{|\nabla\phi|}\cdot\mathbf{n}^{k+1}\right).$$

We concentrate on the second energy $E_\phi(\phi)$ which measures the discrepancy between the regularized normal vector field $\mathbf{n}^{k+1}$ and the normal of the level set from the function ϕ. It is clear that the energy is always positive and is minimized if the normal vector $\mathbf{n}^{k+1}$ pointing the **same** direction as the unit vector $\nabla\phi/|\nabla\phi|$.

In this paper, we follow a similar approach as in the second step of the iterative procedure. We want to determine a curve $\gamma(s)$ such that the tangent is parallel to the given normalized velocity field $\mathbf{u}$, i.e. the normal vector is perpendicular to $\mathbf{u}$ with $\|\mathbf{u}\|=1$. We propose the following variational formulation of finding a curve γ which minimizes

$$E(\gamma)=\int_\gamma\left(\epsilon+\left|(\gamma'(s))^\perp\cdot\mathbf{u}\right|\right)ds.$$

When the curve γ is represented implicitly using the zero level set of a function ϕ, we obtain

$$E(\phi)=\int_\Omega\delta(\phi)|\nabla\phi|(\epsilon+|\mathbf{n}\cdot\mathbf{u}|),$$

where $\mathbf{n}=\nabla\phi/|\nabla\phi|$ is the normal defined along each level curve. The term $|\mathbf{n}\cdot\mathbf{u}|$ enforces that the normal vector on the curve is perpendicular to the vector field $\mathbf{u}$ everywhere. This implies that this closed curve coincides with a certain streamline of $\mathbf{u}$ and hence it is a candidate for limit cycles. The positive regularization parameter ϵ in the functional is chosen to regularize the total length of the curve γ and so the interface will shrink toward the limit cycles if they exist. As a result, given an initial guess (a closed curve), we expect that the zero level set of ϕ will converge to and will coincide with all limit cycles inside.

The optimality condition for minimizing this energy is given by

$$-\delta(\phi)\nabla\cdot\left[\epsilon\frac{\nabla\phi}{|\nabla\phi|}+\left(\frac{\nabla\phi\cdot\mathbf{u}}{|\nabla\phi\cdot\mathbf{u}|}\right)\mathbf{u}\right]=0$$

in the domain Ω with the boundary condition on $\partial\Omega$ given by

$$\left[\epsilon\frac{\nabla\phi}{|\nabla\phi|}+\left(\frac{\nabla\phi\cdot\mathbf{u}}{|\nabla\phi\cdot\mathbf{u}|}\right)\mathbf{u}\right]\cdot\nu=0$$

with ν the outward normal of the boundary. To minimize the energy $E(\phi)$, we apply the variational level set approach in [49] and numerically solve the following gradient descent equation

$$\phi_\tau=-\frac{dE(\phi)}{d\phi}=\epsilon|\nabla\phi|\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}+|\nabla\phi|\nabla\cdot[\mathcal{S}(\nabla\phi\cdot\mathbf{u})\mathbf{u}], \tag{4.1}$$

where $\mathcal{S}(x)=x/|x|$.

Since this gives a nonlinear equation, explicit schemes will not be satisfactory because we need to obtain the steady state solution for any given initial condition. Again we apply the semi-implicit scheme in Section 3 and propose to regularize the evolution by adding and subtracting a curvature term. This gives

$$\phi_\tau=\beta\kappa-\beta\kappa+\epsilon|\nabla\phi|\kappa+|\nabla\phi|\nabla\cdot[\mathcal{S}(\nabla\phi\cdot\mathbf{u})\mathbf{u}].$$

Numerically, we treat the first curvature term implicitly and all other terms on the right-hand side explicitly. Then ϕ^{k+1} is determined by minimizing the functional (3.2) with $\lambda=(\beta\Delta t)^{-1}$ and

$$f(\phi)=\phi+\Delta t\{(\epsilon|\nabla\phi|-\beta)\kappa+|\nabla\phi|\nabla\cdot[\mathcal{S}(\nabla\phi\cdot\mathbf{u})\mathbf{u}]\}.$$

Minimizing the L_2 difference in the update formula in $f(\phi)$, we obtain

$$\beta=\frac{\int_\Omega\kappa|\nabla\phi^k|\left\{\epsilon\kappa+\nabla\cdot\left[\mathcal{S}(\nabla\phi^k\cdot\mathbf{u})\mathbf{u}\right]\right\}d\mathbf{x}}{\int_\Omega\kappa^2 d\mathbf{x}}.$$

In the above construction, note that the method only tries to determine a closed curve that follows a streamline of $\mathbf{u}$. The algorithm does not distinguish if the corresponding limit cycle is a stable one, an unstable one, or neither. Here, we first propose a simple Eulerian method to test if the zero level set gives an unstable limit cycle. Then we will extend the approach for stable cycles.

Let $F_{\tau_1}^{\tau_2}:\mathbb{R}^2\to\mathbb{R}^2$ be the flow map of the velocity field $\mathbf{u}$, i.e. a particle located at $\mathbf{x}$ at time τ_1 will be located at $F_{\tau_1}^{\tau_2}(\mathbf{x})$ at time τ_2. Let $\phi^\infty(\mathbf{x})$ be the steady state solution to equation (4.1) and $\psi(\mathbf{x},\tau)=\phi^\infty(F_\tau^0(\mathbf{x}))$ which denotes the level set value at the take-off location corresponding to its arrival location $\mathbf{x}$ at time τ. Since the value $\psi(\mathbf{x},\tau)$ is constant along a particle trajectory, the material derivative of $\psi(\mathbf{x},\tau)$ equals to zero, i.e.

$$\frac{D\psi(\mathbf{x},\tau)}{D\tau}=0,$$

which implies the following level set equation, or the Liouville equation,

$$\frac{\partial \psi(\mathbf{x},\tau)}{\partial \tau}+(\mathbf{u}\cdot\nabla)\psi(\mathbf{x},\tau)=0 \tag{4.2}$$

with the initial condition $\psi(\mathbf{x},0)=\phi^{\infty}(\mathbf{x})$. If the limit cycle is an unstable one, particles near the zero level set at time $\tau=0$ will flow outward. Therefore, for any $\tau>0$, we will be able to find a tube $\Gamma_\delta=\{|\phi^{\infty}(\mathbf{x})|\le\delta\}$ around the zero level set of $\phi^{\infty}(\mathbf{x})$ so that $|\psi(\mathbf{x},\tau)|<|\psi(\mathbf{x},0)|$ for any $\mathbf{x}\in\Gamma_\delta$. This property provides a nice numerical test for the stability of the limit cycle. We solve the level set equation (4.2) for some $\tau>0$ and try to determine the set Γ_δ for some constant $\delta>0$. If

$$\frac{1}{|\Gamma_\delta|}\int_{\Gamma_\delta}\left|\frac{\psi(\mathbf{x},\tau)}{\psi(\mathbf{x},0)}\right|d\mathbf{x}<1,$$

where $|\Gamma_\delta|=\int_{\Gamma_\delta}d\mathbf{x}$ is the area of the tube Γ_δ, we conclude that the limit cycle is an unstable one.

To check if a limit cycle is a stable one, we consider the following lemma.

Lemma 4.1. Γ *is a stable limit cycle of the planar system* $\mathbf{x}'(t)=\mathbf{u}(\mathbf{x}(t))$ *if and only if it is an unstable limit cycle of the system* $\mathbf{x}'(t)=-\mathbf{u}(\mathbf{x}(t))$.

Therefore, if the limit cycle is stable, it becomes unstable if we reverse the time direction and so we propose to check the quantity

$$\frac{1}{|\Gamma_\delta|}\int_{\Gamma_\delta}\left|\frac{\psi(\mathbf{x},-\tau)}{\psi(\mathbf{x},0)}\right|d\mathbf{x}$$

for some $\delta,\tau>0$. If it is less than 1, we conclude that the limit cycle is a stable one. A summary of this algorithm is given here.

Algorithm 4.1: Classification of the limit cycle

1. Initialize $\psi(\mathbf{x},0)=\phi^{\infty}(\mathbf{x})$. Pick $\delta>0$.
2. Solve the level set equation (4.2) both forward in time and backward in time to obtain $\psi(\mathbf{x},\tau)$ and $\psi(\mathbf{x},-\tau)$.
3. Determine
$$\frac{1}{|\Gamma_\delta|}\int_{\Gamma_\delta}\left|\frac{\psi(\mathbf{x},\tau)}{\psi(\mathbf{x},0)}\right|d\mathbf{x} \text{ and } \frac{1}{|\Gamma_\delta|}\int_{\Gamma_\delta}\left|\frac{\psi(\mathbf{x},-\tau)}{\psi(\mathbf{x},0)}\right|d\mathbf{x}.$$
4. If the first quantity is less than 1, then the limit cycle is an unstable one. If the second quantity is less than 1, then the limit cycle is stable. Otherwise, the test is inconclusive.

Numerically, the level set equation (4.2) can be solved by any well-established robust and high order accurate numerical methods, such as WENO5-TVDRK2 [16, 30, 43]. For moderate to large τ, it is also possible to apply the backward phase flow method developed in [28] in order to obtain a more computational efficient method.

5 Examples

5.1 Motion by mean curvature

In the first example, as a benchmark, we consider the motion by mean curvature. We consider the evolution of a circle under the motion. The exact radius of the solution can be computed explicitly and is given by $r(t)=\sqrt{r_0^2-2t}$, where r_0 is the initial radius of the circle and is chosen to be 0.25. Since the circle shrinks and disappears at $t=0.5r_0^2$, we compute the solution only up to $t=t_f=0.4r_0^2$. The final radius of the circle is given by $r(t_f)=0.11180$.

Fig. 1(a) and (b) show the solutions at various time levels and on various mesh sizes, but with a fixed Δt given by $\Delta t=t_f/32$. The finest solution is computed using a mesh

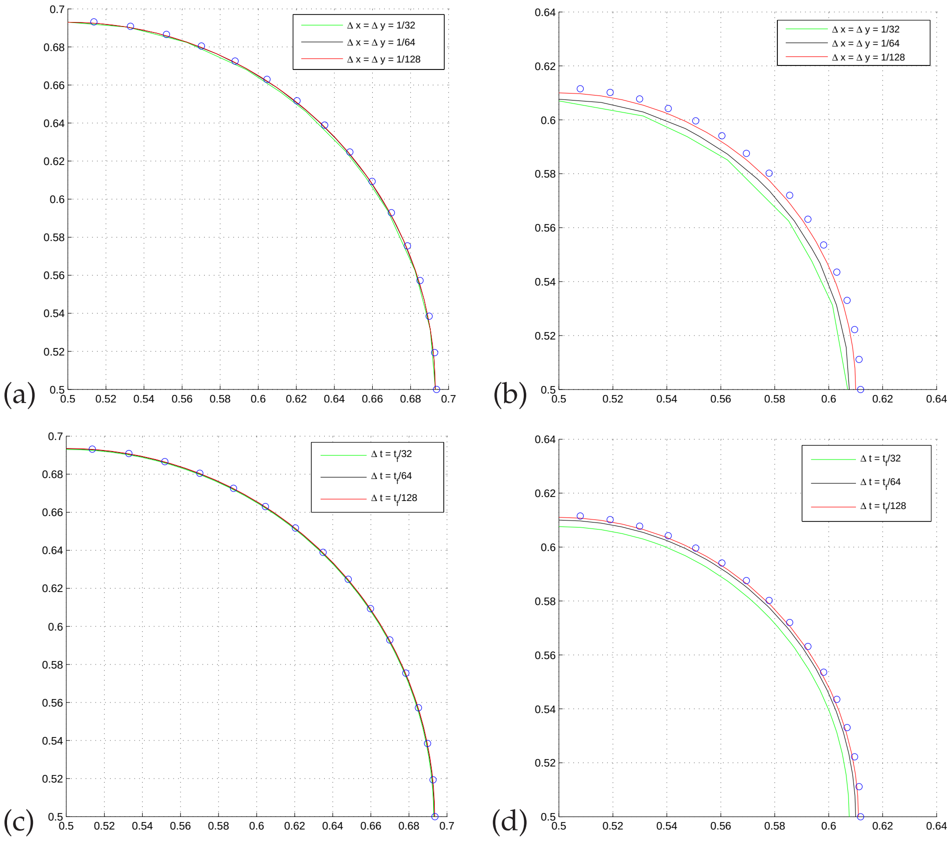

Figure 1: (Example 5.1) Motion by mean curvature of an initial circle of radius $r_0=0.25$ with $t_f=0.4r_0^2$. Solutions at (a) $t=t_f/2$ and (b) $t=t_f$ with a fixed $\Delta t=t_f/32$ and different Δx's. For $\Delta x=1/32$, $\Delta t=0.8\Delta x^2$. For $\Delta x=1/64$, $\Delta t=3.2\Delta x^2$. For $\Delta x=1/128$, $\Delta t=12.8\Delta x^2$. Solutions at (c) $t=t_f/2$ and (d) $t=t_f$ with a fixed $\Delta x=1/128$ and different Δt's. For $\Delta t=t_f/32$, $\Delta t=12.8\Delta x^2$. For $\Delta t=t_f/64$, $\Delta t=6.4\Delta x^2$. For $\Delta t=t_f/128$, $\Delta t=3.2\Delta x^2$. The exact solution is plotted using small blue circles.

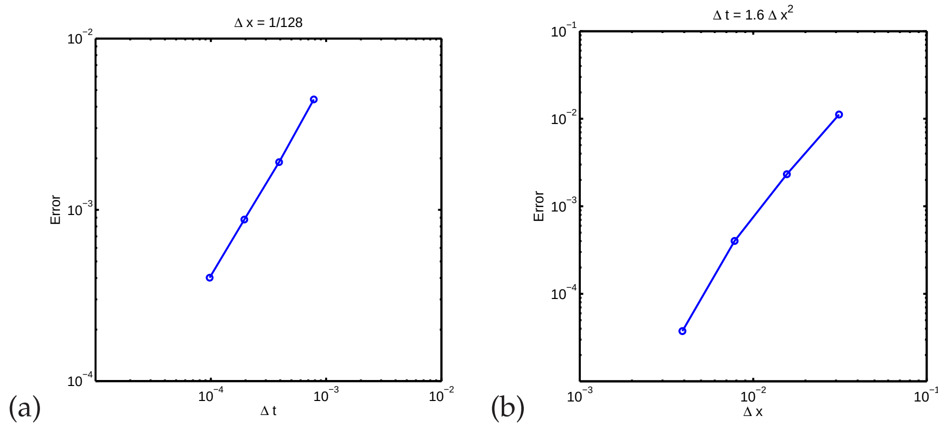

Figure 2: (Example 5.1) Convergence test on the numerical scheme for the motion by mean curvature of an initial circle of radius $r_0 = 0.25$ with $t_f = 0.4r_0^2$.

of size $\Delta x = 1/128$ (equivalent to $\Delta t = 12.8\Delta x^2$) which clearly violates the usual time step restriction for a usual explicit method. In Fig. 1(c) and (d), we demonstrate the convergence behavior for the time refinement. We have fixed the underlying mesh in these figures with $\Delta x = 1/128$ but vary the number of steps marching to the same $t = t_f$. For the least number of time iterations, we choose $\Delta t = t_f/32$ which is equivalent to $\Delta t = 12.8\Delta x^2$. These figures show clearly that Algorithm 3.2 is stable for large time steps which violates the stability condition in typical explicit methods.

To study the order of convergence, we first fix $\Delta x = 1/128$ and vary Δt from $\frac{0.2}{2^8}$ to $\frac{0.2}{2^{11}}$ to calculate the L_∞ error of our numerical result at $t = t_f$ defined by

$$E = \max_\theta |r_{\Delta x}(\theta) - r_{\text{exact}}|.$$

Fig. 2(a) shows the variation of error with respect to the change of Δt. It is plotted after taking logarithms of both quantities and hence the slope of the line denotes the order of convergence with respect to Δt which is approximately 1 (1.1525). In Fig. 2(b), we let $\Delta t = 1.6\Delta x^2$ and vary Δx from $1/32$ to $1/256$ and the slope is about 2 (2.5336).

As a comparison, we have also shown some results by the explicit scheme in Fig. 3. In Fig. 3(a), we have fixed $\Delta x = 1/64$ and have used various Δt's in the computations. We find that the solution becomes unstable when we increase Δt to $t_f/64$ which corresponds to $\Delta t = 1.6\Delta x^2$. In Fig. 3(b), we fix $\Delta x = 1/128$ and, once again, the solution becomes unstable when we increase Δt over $1.6\Delta x^2$. As a result, our semi-implicit scheme has greatly relaxed the time-step restriction compared to the explicit scheme.

5.2 Affine invariant motion by curvature

In this part we study the affine invariant motion by curvature [2,12,34] where the normal velocity depends on the cubic root of the curvature, i.e.

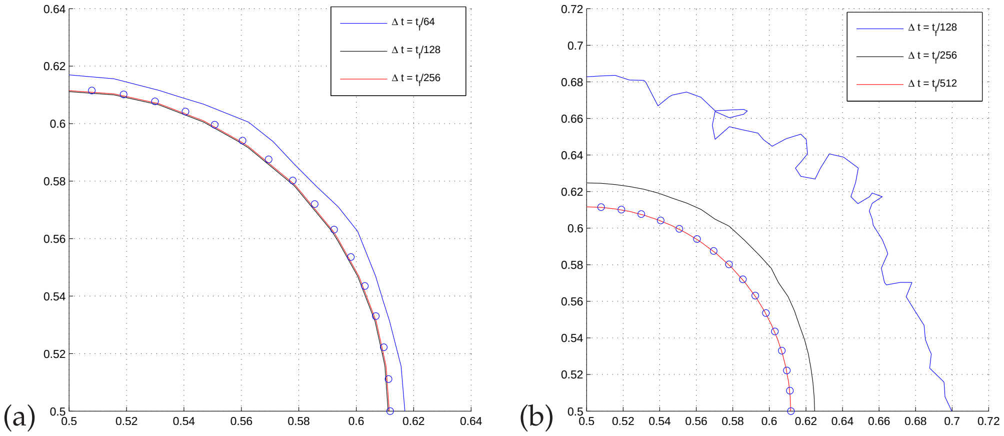

Figure 3: (Example 5.1) Motion by mean curvature of an initial circle of radius $r_0 = 0.25$ with $t_f = 0.4r_0^2$ using the explicit scheme. Solutions at $t = t_f$ with (a) $\Delta x = 1/64$, (b) $\Delta x = 1/128$ and different Δt's. The exact solution is plotted using small blue circles.

$$\phi_t = |\nabla\phi|\kappa^{1/3}.$$

Like in Section 5.1, we first compute the evolution of a circle initially centered at $(0.5,0.5)$ of radius $r_0 = 0.25$. The exact radius of the solution can be computed explicitly and is given by $r(t) = (r_0^{\frac{4}{3}} - \frac{4}{3}t)^{\frac{3}{4}}$. We also compute the solution up to $t_f = 0.4r_0^2 = 0.025$.

Fig. 4(a) shows the solutions on various mesh sizes but with a fixed Δt given by $\Delta t = t_f/32$. In Fig. 4(b) we have fixed the underlying mesh with $\Delta x = 1/128$ but vary the number of steps marching to the same $t = t_f$.

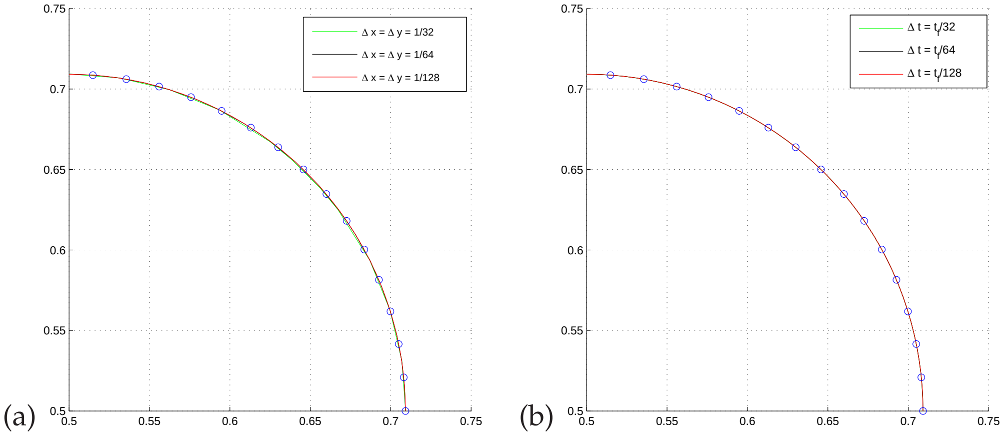

Figure 4: (Example 5.2) Affine invariant motion by curvature of an initial circle of radius $r_0 = 0.25$ with $t_f = 0.4r_0^2$. (a) Solutions at $t = t_f$ with a fixed $\Delta t = t_f/32$ and different Δx's. (b) Solutions at $t = t_f$ with a fixed $\Delta x = 1/128$ and different Δt's. The little blue circles correspond to the exact solution.

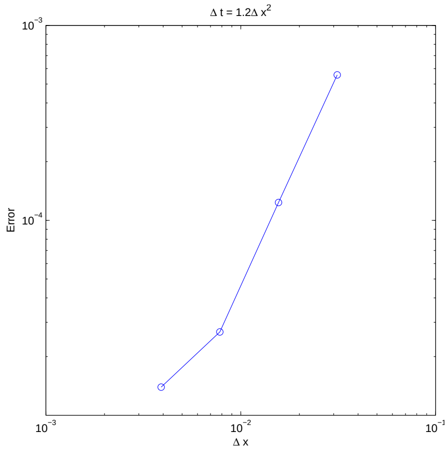

Figure 5: (Example 5.2) Convergence test on the numerical scheme using the affine invariant motion by curvature of an initial circle of radius $r_0 = 0.25$ with $t_f = 0.4r_0^2$.

We have also tested the convergence of our numerical solution in Fig. 5 using $\Delta t = 1.2\Delta x^2$ and different Δx. The slope is about 2 which implies that the error is second order with respect to Δx and first order with respect to Δt.

Another interesting example is the evolution of a 3-folded star shape up to $t = t_f = 7.5\times10^{-2}$. Fig. 6(a) shows the solutions at $t = t_f$ with the same time step $\Delta t = t_f/64$ but different Δx's. We can see that the solutions remain stable even we have taken a large time step of $\Delta t = 76.8\Delta x^2$ which clearly violates the expected stability condition for an

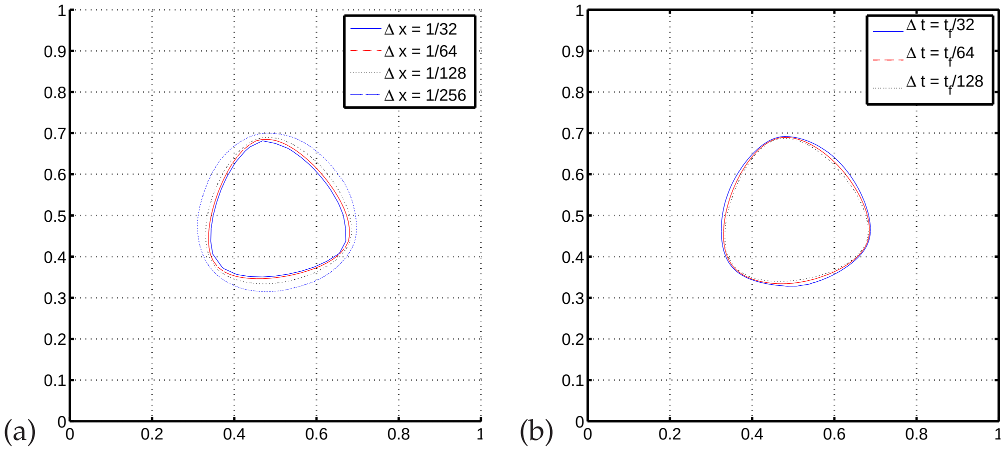

Figure 6: (Example 5.2) Affine invariant motion by curvature of an initial 3-folded star shape with $t_f = 7.5\times10^{-2}$. (a) Solutions at $t = t_f$ with a fixed $\Delta t = t_f/64$ and different Δx's. For $\Delta x = 1/32$, $\Delta t = 1.2\Delta x^2$. For $\Delta x = 1/64$, $\Delta t = 4.8\Delta x^2$. For $\Delta x = 1/128$, $\Delta t = 19.2\Delta x^2$. For $\Delta x = 1/256$, $\Delta t = 76.8\Delta x^2$. (b) Solutions at $t = t_f$ with a fixed $\Delta x = 1/128$ and different Δt's. For $\Delta x = 1/32$, $\Delta t = 38.4\Delta x^2$. For $\Delta x = 1/64$, $\Delta t = 19.2\Delta x^2$. For $\Delta x = 1/128$, $\Delta t = 9.6\Delta x^2$.

explicit scheme. In Fig. 6(b), we fix $\Delta x = 1/128$ and then recompute the solution using different Δt's so that the ratio of Δt to Δx^2 ranges from 9.6 to 38.4. All these solutions do not satisfy the expected stability condition of the explicit scheme.

5.3 Min-curvature flow

In this example, we consider the following min-curvature flow given by

$$\phi_t = |\nabla\phi| \min(0,\kappa).$$

This flow has been proposed for surface regularization in image processing [31]. The function v_n is chosen so that the interface locally stays unmoved if the curvature is positive and it expands locally if the curvature is negative. The motion becomes stationary when the curvature is everywhere non-negative, which gives the convex hull of the initial shape. The algorithm proposed in [12] requires that the normal velocity is an odd, increasing, and Lipschitz function in κ. These requirements make it difficult for that particular scheme to be generalized for this nonlinear evolution.

Fig. 7 considers the evolution of an initial 3-folded star shape at different time levels under this min-curvature flow for time up to $t = t_f = 0.04$. Similar to the previous example, we have tested our algorithm using various Δx and Δt such that the ratio $\Delta t/\Delta x^2$ ranges from 0.32 to 5.12. In Fig. 8 we have listed some of them. For a relatively large Δx or a relatively large Δt, the solution is not accurate at all since the convex region ($\kappa > 0$) of the interface could shrink in time (Fig. 8(a) with $\Delta x = 1/8$) and the concave region ($\kappa < 0$) of the interface could over-expand and form a convex region (Fig. 8(b) with $\Delta t = t_f/32$). However, we do not see any instability in the evolution. The time step is chosen not according to the stability condition ($\mathcal{O}(\Delta x^2)$), but according to the accuracy consideration.

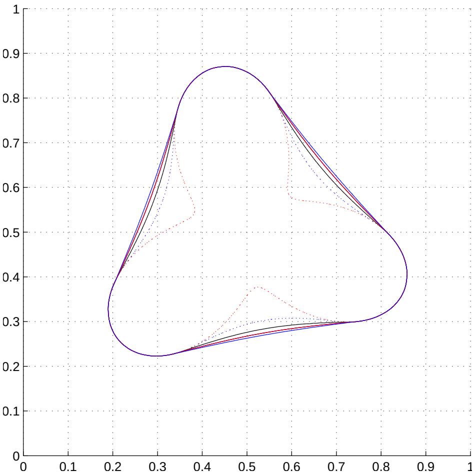

Figure 7: (Example 5.3) Min-Curvature flow of an initial 3-folded star shape with $t_f = 0.04$ using $\Delta x = 1/128$ and $\Delta t = t_f/512$. Evolution of the interface at $t = 0$, $t_f/4$, $t_f/2$, $3t_f/4$, and t_f plotted in dashed red, dashed blue, solid black, solid red and solid blue, respectively.

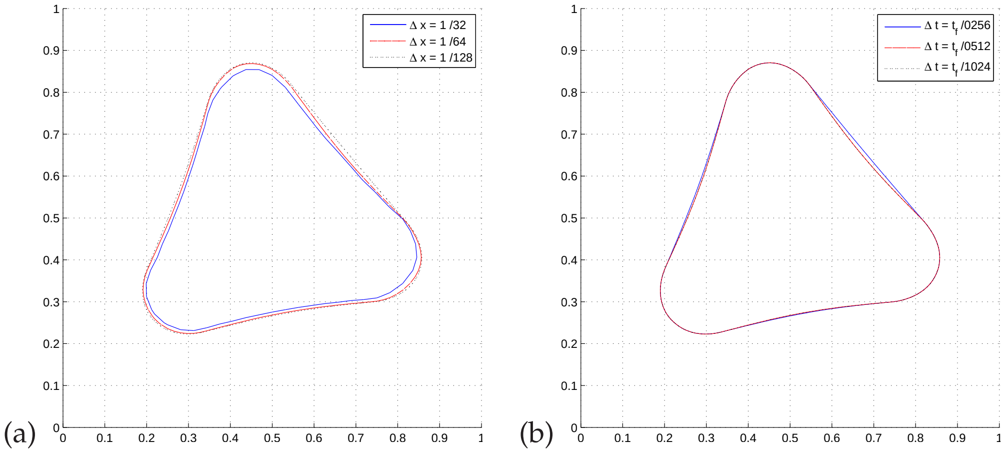

Figure 8: (Example 5.3) Min-Curvature flow of an initial 3-folded star shape with $t_f=0.04$. (a) Solutions at $t=t_f$ with a fixed $\Delta t=t_f/128$ and different Δx's. For $\Delta x=1/32$, $\Delta t=0.32\Delta x^2$. For $\Delta x=1/64$, $\Delta t=1.28\Delta x^2$. For $\Delta x=1/128$, $\Delta t=5.12\Delta x^2$. (b) Solutions at $t=t_f$ with a fixed $\Delta x=1/128$ and different Δt's. For $\Delta t=t_f/256$, $\Delta t=2.56\Delta x^2$. For $\Delta t=t_f/512$, $\Delta t=1.28\Delta x^2$. For $\Delta t=t_f/1024$, $\Delta t=0.64\Delta x^2$.

5.4 Computational time

To further look at the computational efficiency of the proposed numerical approach, we compare the computational (CPU) time of the explicit scheme (2.2), the simple semi-implicit scheme (2.3) and the proposed numerical approach. The numerical results are obtained using a laptop computer with a 2.5 GHz Intel core i7 processor. Fig. 9(a) shows the computational time for solving the motion by mean curvature on a fixed $\Delta x=1/128$ but various Δt's. The blue solid line on the bottom is the computational time (in second) for the simple explicit scheme. For $\Delta t > t_f/256$, the solution becomes unstable and so we do not report the corresponding computational time here. The computational times for the simple semi-implicit scheme (2.3) for different Δt's are shown using the red dash line. The sparse matrix at each time step is constructed using the MATLAB sparse data structure and is inverted using the backslash operator rather than any specific iterative solver. The computational times for each iteration of both schemes depend only on the mesh size, which implies that the CPU time is inversely proportional to Δt. This explains why the curve plotted in the log-log scale has slope approximately -1.

The computational time for the proposed scheme is shown using the black dash dot line, in the middle of two curves. At each intermediate step, we propose to solve one ROF functional to update the level set function using an iterative scheme. We agree that it is indeed more expensive than an explicit update using (2.2) since iterative scheme has to be used to minimize the ROF functional. In the current paper we are following the first order primal-dual algorithm [9] which might not be the latest algorithm in the field. Nevertheless, the proposed scheme is already faster than the simple semi-implicit scheme (2.3). We expect that the proposed method can definitely be further improved

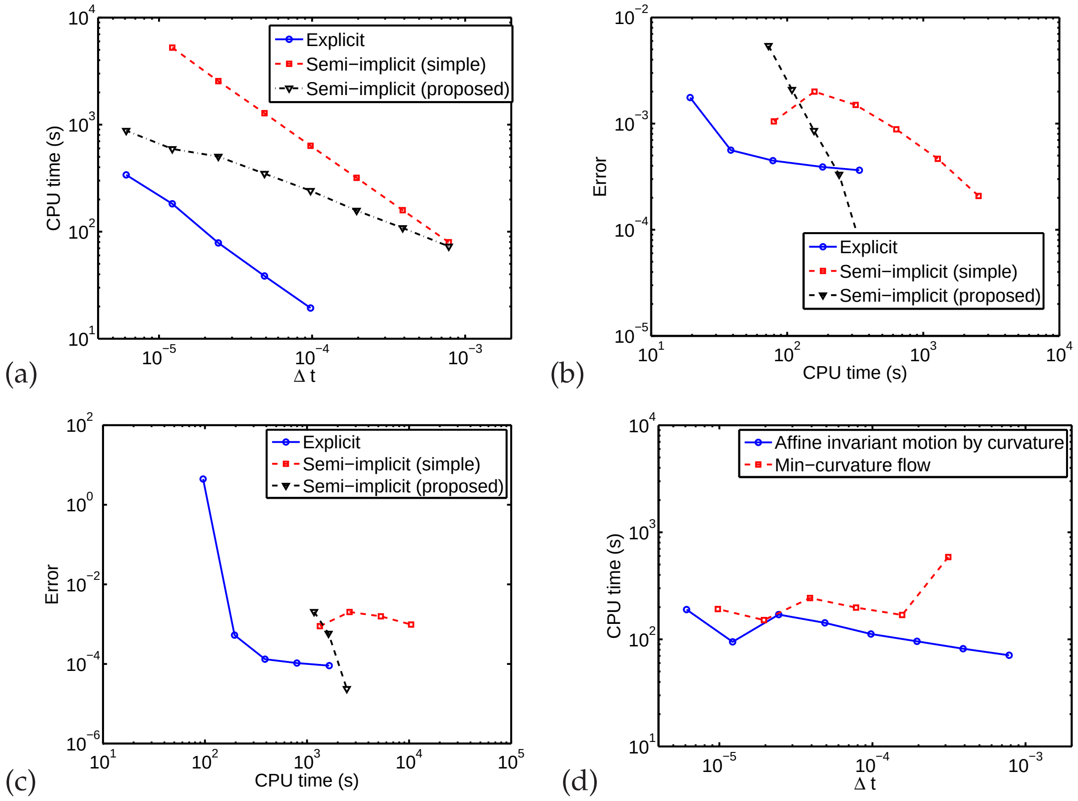

Figure 9: (a) Total computational (CPU) time (in second) for the motion by mean curvature of the explicit scheme (2.2), the simple semi-implicit scheme (2.3) and the proposed numerical approach vs Δt using $\Delta x = 1/128$. (b) Error in the final solution vs the total computational (CPU) time for various schemes using $\Delta x = 1/128$. (c) Error in the final solution vs the total computational (CPU) time for various schemes using $\Delta x = 1/256$. (d) Computational times (in second) for the affine invariant motion by curvature and the min-curvature flow using the proposed numerical regularization approach.

later as the computational time of the ROF solver is further shortened. More importantly, the computational time for the proposed algorithm does not increase linearly like the explicit scheme and the simple semi-implicit scheme. The average CPU time for each time iteration decreases as we decrease Δt. This is because the level set function ϕ^{k+1} is more similar to ϕ^k for smaller Δt, which implies that the number of iterations required to minimize the ROF function is reduced. Therefore, the CPU time for each iteration is reduced.

Another way to consider the efficiency of the proposed algorithm is to look at the amount of computational time we spent to achieve certain accuracy in the solution. In Fig. 9(b), we plot the infinity-error in the location of the zero level set versus the CPU time for the data we have collected in (a). In order to achieve an accuracy of approximately 4×10^{-4}, we found that the time we spent on the proposed scheme (using the current ROF solver) is already shorter than that of the explicit scheme, and is also one order magnitude faster than that using the simple semi-implicit scheme. However, we have to be careful interpreting these data when it involves measuring errors in the solution. In particular, since the error in the solution is expected to be of order $\mathcal{O}(\Delta t+\Delta x^2)$ and these data are obtained on a *fixed* mesh $\Delta x=1/128$, we do not expect errors from the scheme can be reduced indefinitely as we refine Δt, i.e. increase the CPU time. This explains for example why the blue solid line is not a straight line of slope -1. In Fig. 9(c), we have repeated the calculations but on a mesh $\Delta x=1/256$ and have obtained some similar trends in the behavior of the numerical solutions.

We have also collected the computational times for the affine invariant motion by curvature (the blue solid line) and the min-curvature flow (the red dash line), as shown in Fig. 9(d). Similar to the motion by mean curvature, the computational time for the affine invariant motion by curvature shows the same trend. As we decrease Δt (while fixing the final time), the CPU time increases but still slower than $1/\Delta t$. Concerning the min-curvature flow, on the other hand, we find that for a relatively large Δt, the overall computational time is longer. This is because the change in the solution is significant and, therefore, the current ROF solver requires more iterations to achieve the required accuracy.

5.5 Application to limit cycles extraction

In this section, we will apply the fast semi-implicit scheme to extract limit cycles of different planar dynamical systems as described in Section 4. The first example is taken from [32] where the dynamical system is given by

$$\begin{aligned} \dot{x_1} &= -x_2+x_1(x_1^2+x_2^2-1), \\ \dot{x_2} &= x_1+x_2(x_1^2+x_2^2-1). \end{aligned} \tag{5.1}$$

It can be determined that $x_1^2+x_2^2=1$ is a stable limit cycle by using the polar coordinates transform. In this example, we use a circle of radius 1.5 as our initial condition. As shown in Fig. 10, our numerical approach gives a solution (blue curve) which matches very well with the exact solution (red curve).

Another example is the Van der Pol Oscillator and the corresponding dynamical system is given by

$$\begin{aligned} \dot{x_1} &= kx_2, \\ \dot{x_2} &= -kx_1+\epsilon(1-(kx_1)^2)kx_2, \end{aligned} \tag{5.2}$$

where k is used to scale the axis and ϵ is a parameter which determines the shape of the limit cycle. Fig. 11 shows the numerical solution to (5.2) with (a) $k=3$, $\epsilon=0.2$, (b) $k=4$, $\epsilon=0.5$. In our numerical implementation, we have chosen $\Delta x=1/64$, $\Delta t=0.8\Delta x^2$ for Fig. 11(a) and $\Delta x=1/32$, $\Delta t=2\Delta x^2$ for Fig. 11(b). As we can see, both of them match very well with exact locations of the limit cycles.

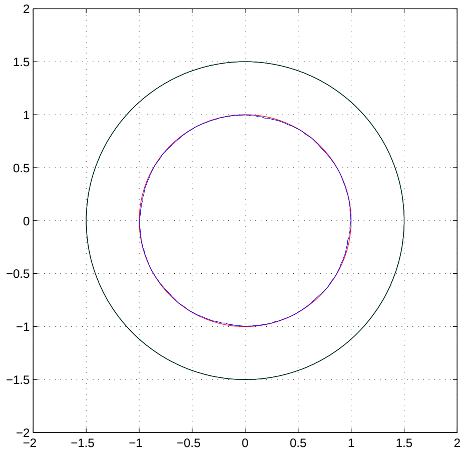

Figure 10: (Example 5.5) Numerical approximation to the limit cycle of (5.1). The blue closed curve is the numerical approximation obtained by our proposed scheme with $\Delta x = 1/32$ and $\Delta t = 1.5\Delta x^2$. The black one is the initial guess and the red one is the exact location of the limit cycle. This result is obtained after only a few hundred iterations.

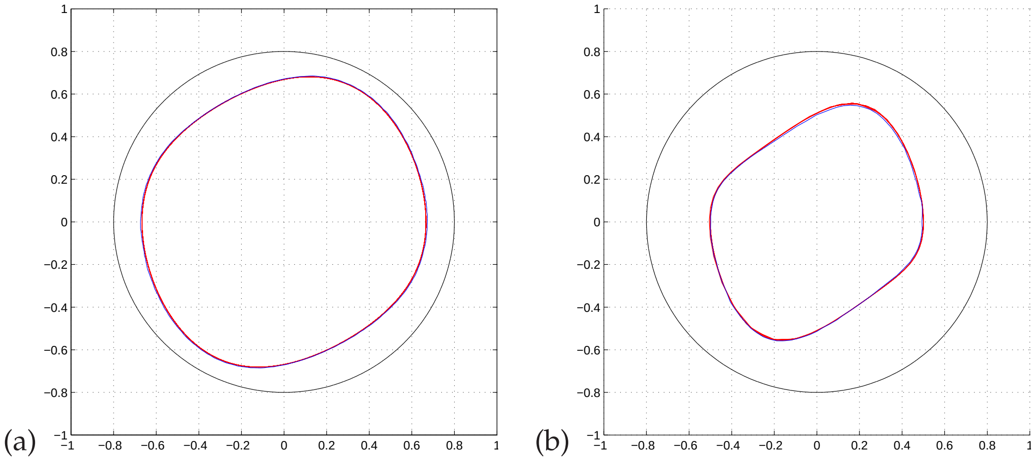

Figure 11: (Example 5.5) Numerical approximation to the limit cycle of (5.2) with (a) $k=3$, $\epsilon=0.2$, (b) $k=4$, $\epsilon=0.5$. The blue closed curve is our numerical approximation, the black one is the initial guess and the red one is the exact location of the limit cycle.

As we have demonstrated in Section 4, our semi-implicit scheme can also deal with cases with multiple limit cycles. To check this point, we have artificially constructed an example with three limit cycles based on system (5.2). Since system (5.2) has a unique limit cycle within the domain $[-1,1]\times[-3,3]$ for $k=6$ and $\epsilon=0.2$, we propose to periodically extend this system horizontally to the computational domain $[-3,3]\times[-3,3]$ which will consequently contain three congruent limit cycles. Several intermediate evolutions of the interface are shown in Fig. 12(a)-(e). In our implementation, we choose $\Delta x = 3/64$, $\Delta t = 5\Delta x^2$, and the steady state solution can be obtained after a few thousand iterations.

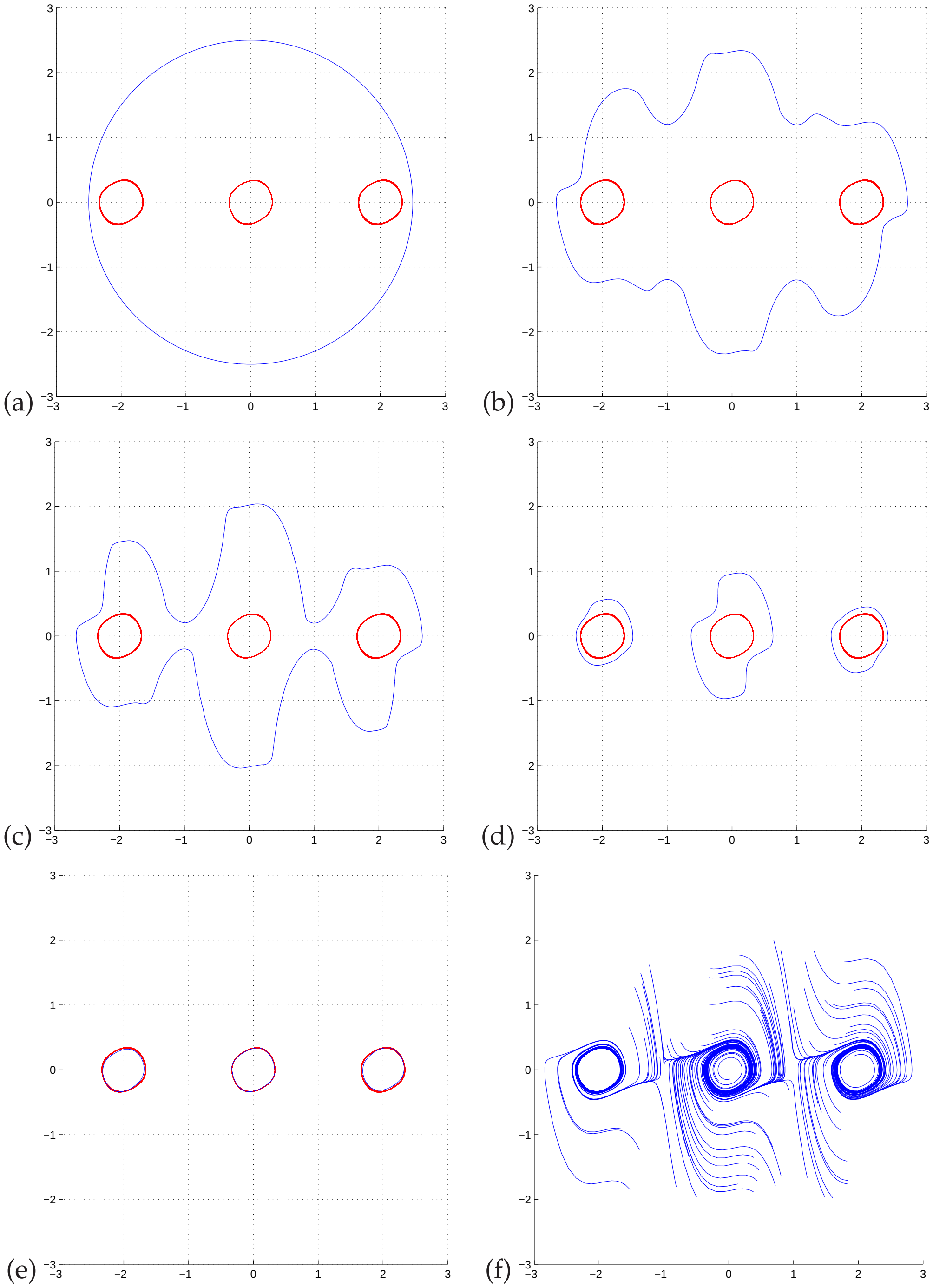

Figure 12: (Example 5.5) Numerical approximation to the three limit cycles of the system obtained by periodically extending system (5.2) horizontally with $k=6$, $\epsilon=0.2$. (a-e) The blue curves show the evolution of the interface and the red ones are the three limit cycles. (f) Trajectories of several Lagrangian particles.

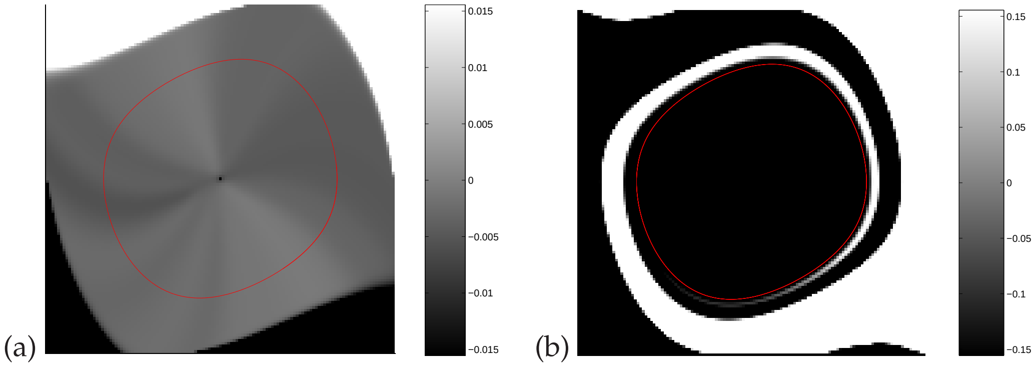

Figure 13: (Example 5.5) Solutions to the level set equation (4.2). (a) $\psi(\mathbf{x},-20)$, (b) $\psi(\mathbf{x},20)$. The limit cycle is also plotted on top of the solutions using a solid red line.

As a comparison, we randomly choose one hundred initial locations and plot their trajectories from $t=0$ to $t=100$ in Fig. 12(f). In practice, it is not straight-forward to determine all limit cycles within a finite computational domain in the typical Lagrangian framework. One might need to shoot out many rays in order to have a better confidence in capturing all stable limit cycles. In our variational framework, the evolution of the zero level set will be able to capture multiple disjoint limit cycles at once.

To further classify the stability of the limit cycle, we follow Algorithm 4.1 to obtain the solution to the level set equation (4.2). We consider the example corresponding to Fig. 11(a). In the figure, the blue closed curve is indeed the zero level contour of the steady state solution $\phi^\infty(\mathbf{x})$. In this case, we solve Eq. (4.2) both backward in time to $\tau=-20$ and forward in time to $\tau=20$, respectively, with the initial condition $\psi(\mathbf{x},0)=\phi^\infty(\mathbf{x})$. The solutions $\psi(\mathbf{x},-20)$ and $\psi(\mathbf{x},20)$ are shown in Fig. 13(a) and (b), respectively. Fig. 13(a) shows that $|\psi(\mathbf{x},-20)|$ is less than $\Delta x=3/64$ in the middle square-shaped bulk, while Fig. 13(b) shows that $|\psi(\mathbf{x},20)|$ is greater than approximately $10\Delta x$ for almost any $\mathbf{x}$ in the whole computational domain. As a result, we can conclude that the limit cycle in Fig. 11(a) is a stable one. We can also obtain the same conclusion from our Algorithm 4.1. For example, we have picked $\delta=3\Delta x$ and we have

$$\int_{\Gamma_\delta}\left|\frac{\psi(\mathbf{x},-20)}{\psi(\mathbf{x},0)}\right|d\mathbf{x}=0.1937 \quad \text{and} \quad \int_{\Gamma_\delta}\left|\frac{\psi(\mathbf{x},20)}{\psi(\mathbf{x},0)}\right|d\mathbf{x}=20.7458$$

while $|\Gamma_\delta|=0.3982$. As a result, we have $\frac{1}{|\Gamma_\delta|}\int_{\Gamma_\delta}\left|\frac{\psi(\mathbf{x},-20)}{\psi(\mathbf{x},0)}\right|d\mathbf{x}<1$ and hence the limit cycle in Fig. 11(a) is a stable one according to Algorithm 4.1.

6 Conclusion

In this paper, we have proposed a new semi-implicit scheme to compute a class of curvature dependent flows, which is developed based on adding-and-subtracting a *curvature* term. This scheme extends a previous approach proposed in [7, 8], while it is still simple enough to be extended to a much wider class of curvature dependent motions. Because the proposed algorithm is derived based on the semi-implicit time discretization, the time-step size can be greatly relaxed comparing to the typical explicit scheme, i.e. $\Delta t \gg \Delta x^2$. As an interesting and important application, we have applied our scheme to extract limit cycles of planar dynamical systems. We have demonstrated that the method can automatically extract multiple disconnected limit cycles inside a computational domain. Future extensions include applying to higher order geometrical motions such as the motion by surface diffusion and the Willmore flow.

Acknowledgments

The work of Leung was supported in part by the RGC under Grant 605612.

References

[1] F. Almgren, J.E. Taylor, and L. Wang. Curvature-driven flow: A variational approach. SIAM J. Control and Optimization, 31:387–437, 1993.

[2] S. Angenent, G. Sapiro, and A. Tannenbaum. On the affine invariant heat equation for nonconvex curves. Journal of the American Mathematical Society, 11:601–634, 1998.

[3] K.J. Arrow, L. Hurwics, and H. Uzawa. Studies in linear and non-linear programming. Stanford Mathematical Studies in the Social Sciences, Vol II, Stanford University Press, 1958.

[4] G. Aubert and P. Kornprobst. Mathematical Problems in Image Processing - Partial Differential Equations and the Calculus of Variations. Springer, 2006.

[5] G. Barles and C. Georgelin. A simple proof of convergence for an approximation scheme for computing motions by mean curvature. SIAM J. Num. Anal., 32:484–500, 1995.

[6] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. International Journal of Computer Vision, 22(1):61–79, 1997.

[7] A. Chambolle. An algorithm for mean curvature motion. Interfaces and Free Boundaries, 6:195–218, 2004.

[8] A. Chambolle. An algorithm for total variation minimization and applications. Journal of Mathematical Imaging and Vision, 20:89–97, 2004.

[9] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. Journal of Mathematical Imaging and Vision, 40(1):120–145, 2011.

[10] T. Chan and J. Shen. Image Processing and Analysis: Variational, PDE, Wavelet and Stochastic Methods. SIAM, Pennsylvania, US., 2005.

[11] J. Esch and T.D. Rogers. On the detection of limit cycles by the variational velocity method. Acta Applicandae Mathematicae, 54:345–354, 1998.

[12] S. Esedoglu, S. Ruuth, and R. Tsai. Diffusion generated motion using signed distance functions. J. Comput. Phys., 229:1017–1042, 2010.

[13] E. Esser, X. Zhang, and T. Chan. A general framework for a class of first order primal-dual algorithms for tv minimization. UCLA CAM 09-67, 2009.

[14] D. Goldfarb and W. Yin. Second-order cone programming methods for total variation based image restoration. SIAM J. on Scientific Computing, 27:622–645, 2005.

[15] T. Goldstein and S. Osher. The split bregman method for l1 regularized problems. SIAM J. Imaging Sciences, 2:323–343, 2008.

[16] S. Gottlieb and C.-W. Shu. Total variation diminishing Runge-Kutta schemes. Mathematics of Computation, 67:73–85, 1998.

[17] E.T. Hale, W. Yin, and Y. Zhang. Fixed-point continuation for l1-minimization: Methodology and convergence. SIAM J. on Optimization, 19:1107–1130, 2008.

[18] G. Haller. Distinguished material surfaces and coherent structures in three-dimensional fluid flows. Physica D, 149:248–277, 2001.

[19] G. Haller and G. Yuan. Lagrangian coherent structures and mixing in two-dimensional turbulence. Physica D, 147:352–370, 2000.

[20] Y. He, Y. Liu, and T. Tang. On large time-stepping methods for the cahn-hilliard equation. Appl. Num. Math., 57:616–628, 2007.

[21] D. Hilbert. Mathematical problems. Bull. Amer. Math. Soc., 8:437–479, 1902.

[22] Y. Ilyashenko. Centennial history of Hilbert's 16th problem. Bull. Amer. Math. Soc., 39(3):301–354, 2002.

[23] G. S. Jiang and D. Peng. Weighted ENO schemes for Hamilton-Jacobi equations. SIAM J. Sci. Comput., 21:2126–2143, 2000.

[24] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. International Journal of Computer Vision, 1(4):321–331, 1988.

[25] M. Kimura and H. Notsu. A level set method using the signed distance function. Japan J. Indust. Appl. Math., 19:415–446, 2002.

[26] G.M. Korpelevich. Extrapolational gradient methods and their connection with modified Lagrangians. Ehkon. Mat. Metody., 19:694–703, 1983.

[27] S. Leung. An Eulerian approach for computing the finite time Lyapunov exponent. J. Comput. Phys., 230:3500–3524, 2011.

[28] S. Leung. A backward phase flow method for the finite time Lyapunov exponent. Chaos, 23(043132), 2013.

[29] S. Leung and J. Qian. The backward phase flow and FBI-transform-based Eulerian Gaussian beams for the Schrödinger equation. J. Comput. Phys., 229:8888–8917, 2010.

[30] X. D. Liu, S. J. Osher, and T. Chan. Weighted Essentially NonOscillatory schemes. J. Comput. Phys., 115:200–212, 1994.

[31] R. Malladi and J.A. Sethian. Flows under min/max curvature flow and mean curvature: Applications in image processing. Lecture Notes in Computer Science, 1064, 1996.

[32] D. Mediana and P. Padilla. A geometric approach to invariant sets for dynamical systems. Electronic Journal of Differential Equations, Conference 18:45–56, 2010.

[33] B. Merriman, J. K. Bence, and S. J. Osher. Motion of multiple junctions: a level set approach. J. Comput. Phys., 112:334–363, 1994.

[34] L. Moisan. Affine plane curve evolution: A fully consistent scheme. IEEE Transactions on Image Processing, 7:411–420, 1998.

[35] A. Oberman, S. Osher, R. Takei, and R. Tsai. Numerical methods for smooth and crystalline mean curvature flow. UCLA CAM 10-21, 2010.

[36] S. Osher, Y. Mao, B. Dong, and W Yin. Fast linearized bregman iteration for compressive sensing and sparse denoising. Commun. Math. Sci., 8:93–111, 2010.

[37] S. J. Osher and R. P. Fedkiw. Level Set Methods and Dynamic Implicit Surfaces. Springer-Verlag, New York, 2003.

[38] S. J. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. J. Comput. Phys., 79:12–49, 1988.

[39] T. Pock, D. Cremers, H. Bischof, and A. Chambolle. An algorithm for minimizing the mumford-shah functional. ICCV Proceedings, LNCS, Springer, 2009.

[40] L. Rudin, S.J. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. Physica D, 60:259–268, 1992.

[41] J. Sethian. Curvature and the evolution of fronts. Commun. Math. Phys., 101:487–499, 1985.

[42] S.C. Shadden, F. Lekien, and J.E. Marsden. Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows. Physica D, 212:271–304, 2005.

[43] C. W. Shu. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In B. Cockburn, C. Johnson, C.W. Shu, and E. Tadmor, editors, Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, volume 1697, pages 325–432. Springer, 1998. Lecture Notes in Mathematics.

[44] P. Smereka. Semi-implicit level set methods for curvature and surface diffusion motion. J. Sci. Comput., 1-3:439–456, 2003.

[45] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher. Geometric surface processing via anisotropic diffusion of normals. Proceedings of IEEE Visualization 2002, pages 125–132, 2002.

[46] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher. Geometric surface processing via normal maps. ACM Transactions on Graphics, 22, 2003.

[47] G. You and S. Leung. An Eulerian method for computing the coherent ergodic partition of continuous dynamical systems. J. Comp. Phys., 264:112–132, 2014.

[48] G. You and S. Leung. VIALS: An Eulerian tool based on total variation and the level set method for studying dynamical systems. J. Comp. Phys., 266:139–160, 2014.

[49] H.-K. Zhao, T. Chan, B. Merriman, and S. J. Osher. A variational level set approach for multiphase motion. J. Comput. Phys., 127:179–195, 1996.

[50] M. Zhu and T. Chan. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. UCLA CAM 08-34, 2009.