

A second-order in time, BGN-based parametric finite element method for geometric flows of curves

Wei Jiang^a, Chunmei Su^{b,*}, Ganghui Zhang^b

^a School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China

^b Yau Mathematical Sciences Center, Tsinghua University, Beijing, 100084, China

ARTICLE INFO

Keywords:

Parametric finite element method
Geometric flow
Shape metrics
BGN scheme
High-order in time
Energy stability

ABSTRACT

Over the last two decades, the field of geometric curve evolutions has attracted significant attention from scientific computing. One of the most popular numerical methods for solving geometric flows is the so-called BGN scheme, which was proposed by Barrett et al. (2007) [8], due to its favorable properties (e.g., its computational efficiency and the good mesh property). However, the BGN scheme is limited to first-order accuracy in time, and how to develop a higher-order numerical scheme is challenging. In this paper, we propose a fully discrete, temporal second-order parametric finite element method, which integrates with two different mesh regularization techniques, for solving geometric flows of curves. The scheme is constructed based on the BGN formulation and a semi-implicit Crank-Nicolson leap-frog time stepping discretization as well as a linear finite element approximation in space. More importantly, we point out that the shape metrics, such as manifold distance and Hausdorff distance, instead of function norms, should be employed to measure numerical errors. Extensive numerical experiments demonstrate that the proposed BGN-based scheme is second-order accurate in time in terms of shape metrics. Moreover, by employing the classical BGN scheme as mesh regularization techniques, our proposed second-order schemes exhibit good properties with respect to the mesh distribution. In addition, an unconditional interlaced energy stability property is obtained for one of the mesh regularization techniques.

1. Introduction

Geometric flows, which describe the evolution of curves or surfaces over time based on the principle that the shape changes according to its underlying geometric properties, such as the curvature, have been extensively studied in the fields of computational geometry and geometric analysis. In particular, second-order (e.g., mean curvature flow, which is also called as curve-shortening flow for curve evolution) and fourth-order (e.g., surface diffusion flow) geometric flows have attracted considerable interest due to their wide-ranging applications in materials science [6,30], image processing [1], multiphase fluids [20] and cell biology [11]. For more in-depth information, readers can refer to the recent review articles [13,16], and references provided therein.

In this paper, we focus on three different types of geometric flows of curves: curve-shortening flow (CSF), area-preserving curve-shortening flow (AP-CSF) and surface diffusion flow (SDF). First, assume that $\Gamma(t)$ is a family of simple closed curves in the two-dimensional plane. We consider that the curve is governed by the three geometric flows, i.e., its velocity is respectively given by

* Corresponding author.

E-mail addresses: jiangwei1007@whu.edu.cn (W. Jiang), sucm@tsinghua.edu.cn (C. Su), gh-zhang19@mails.tsinghua.edu.cn (G. Zhang).

$$\mathcal{V} = \begin{cases} -\kappa \mathbf{n}, & \text{CSF,} \\ (-\kappa + \langle \kappa \rangle) \mathbf{n}, & \text{AP-CSF,} \\ (\partial_{ss} \kappa) \mathbf{n}, & \text{SDF,} \end{cases} \tag{1.1}$$

where κ is the curvature of the curve, s is the arc-length, $\langle \kappa \rangle := \int_{\Gamma(t)} \kappa ds / \int_{\Gamma(t)} 1 ds$ is the average curvature and \mathbf{n} is the outward unit normal to Γ . Here, we use the sign convention that a unit circle has a positive constant curvature.

By representing the curves $\Gamma(t)$ as a parametrization $\mathbf{X}(\cdot, t) : \mathbb{I} \rightarrow \mathbb{R}^2$, where $\mathbb{I} := \mathbb{R}/\mathbb{Z}$ is the ‘‘periodic’’ interval $[0, 1]$, Barrett, Garcke and Nürnberg [8,13] creatively reformulated the above equations (1.1) into the following coupled forms:

$$\begin{aligned} \partial_t \mathbf{X} \cdot \mathbf{n} &= \begin{cases} -\kappa, & \text{CSF,} \\ -\kappa + \langle \kappa \rangle, & \text{AP-CSF,} \\ \partial_{ss} \kappa, & \text{SDF,} \end{cases} \\ \kappa \mathbf{n} &= -\partial_{ss} \mathbf{X}. \end{aligned} \tag{1.2}$$

Based on the above equations and the corresponding weak formulations, a series of numerical schemes (the so-called BGN schemes) were proposed for solving different geometric flows, such as mean curvature flow and surface diffusion [8,9], Willmore flow [11], anisotropic geometric flow [5], solid-state dewetting [6,30] and geometric flow for surface evolution [10]. Recently, based on the BGN formulation (1.2), structure-preserving schemes have been proposed for axisymmetric geometric equations [4] and surface diffusion [5,7], respectively. In practical simulations, ample numerical results have demonstrated the high performance of the BGN scheme, due to inheriting the variational structure of the original problem and introducing an appropriate tangential velocity to help mesh points maintain a good distribution. However, for the original BGN scheme, because its formal truncation error is $\mathcal{O}(\tau)$, where τ is the time step size, the temporal convergence order of the scheme is limited to the first-order. This has been confirmed by extensive numerical experiments [6–9]. Therefore, how to design a temporal high-order scheme which is based on the BGN formulation (1.2) is challenging and still open. It is also worth noting that rigorous numerical analysis for BGN schemes remains an open problem [13].

In this paper, based on the BGN formulation (1.2), we propose a novel temporal second-order parametric finite element method for solving geometric flows of curves, i.e., CSF, AP-CSF and SDF. Specifically, to discretize the same continuous-in-time semi-discrete formulation as the classical BGN scheme [8], we begin by fixing the unit normal as that on the current curve Γ^m and then discretize other terms using the Crank-Nicolson leap-frog scheme [21]. The resulting scheme is a second-order semi-implicit scheme, which only requires solving a system of linear algebraic equations at each time step. Furthermore, the well-posedness of the fully discrete scheme can be established under suitable assumption conditions. Numerical results have demonstrated that the proposed scheme achieves second-order accuracy in time, as measured by the shape metrics, outperforming the classical BGN scheme in terms of accuracy and efficiency.

It is worth mentioning that there exist several temporal higher-order numerical schemes based on other formulations which have been proposed for simulating geometric flows. For the specific case of curve-shortening flow, a Crank-Nicolson-type scheme combined with tangential redistribution [3] and an adaptive moving mesh method [27] have been developed. Both of the schemes are convergent quadratically in time and fully implicit, requiring to solve a system of nonlinear equations at each time step. Recently, an evolving surface finite element method together with linearly implicit backward difference formulae for time integration for simulating the mean curvature flow has been proposed in [25,26]. In comparison to these existing approaches, our newly proposed scheme is based on the BGN formulation (1.2), then it inherits the variational structure of the original geometric flows, and has very good property with respect to mesh distribution. The new scheme exhibits comparable computational cost to the classical BGN scheme while surpassing it in terms of accuracy. Furthermore, it can be extended easily to other geometric flows with applications to various fields.

The main reason why we have successfully proposed a temporal high-order, BGN-based parametric finite element method for solving geometric flows lies in the following two key points: (1). we choose an appropriate metric (i.e., shape metrics) to measure numerical errors of the proposed schemes; (2). we use the classical first-order BGN scheme as ‘‘a good partner’’ of the proposed scheme to help mesh points maintain a good distribution without sacrificing the accuracy.

How to measure the errors of numerical solutions for geometric flows is an important issue. A natural approach is to use classical Sobolev norms, such as L^2 -norm, L^∞ -norm or H^1 -norm, which are widely used in the numerical analysis for geometric flows [17,18,25,26]. However, when it comes to numerical schemes that involve in tangential movements, these function norms may not be suitable for quantifying the differences between two curves/surfaces. To address this issue, we consider an alternative approach using shape metrics, such as manifold distance (as used in [7,31]) and Hausdorff distance [2]. These metrics provide a measure of how similar or different two curves/surfaces are in terms of their shape characteristics. Extensive numerical experiments have been conducted, and the results demonstrate that our proposed scheme achieves second-order accuracy when measured using shape metrics.

On the other hand, the quality of mesh distribution is always a major concern when simulating geometric flows using parametric finite element methods. It is important to note that the original flow (1.1) requires the curve to evolve only in the normal direction, thus the numerical methods based on (1.1) which prevent tangential movement of mesh points might lead to mesh distortion or clustering during the evolution. To address this issue, various approaches have been proposed in the literature to maintain good mesh quality, e.g., artificial mesh regularization method [14], reparametrization by introducing a tangential velocity [15,19,24,28,29]. On the contrary, the BGN formulation (1.2) does not enforce any condition on the tangential velocity, which allows for an intrinsic tangential motion of mesh points, as demonstrated by the standard BGN scheme [8,9] constructed based on this formulation (1.2). Though the semi-discrete scheme of (1.2), where only spatial discretization is performed, results in precise equidistribution of

mesh points, our proposed fully discrete second-order BGN-based scheme exhibits oscillations in terms of mesh ratio and other geometric quantities, which may lead to instability in certain situations. To address this issue, we implement two classical first-order BGN schemes as mesh regularization procedures to enhance the quality of the mesh. More specifically, (1). we utilize the classical semi-implicit BGN scheme when poorly distributed polygonal approximations are detected. Extensive numerical experiments have shown that this approach improves the stability of the new scheme and significantly enhances the mesh quality. Importantly, numerous numerical experiments have also demonstrated that this mesh regularization only occurs infrequently throughout the evolution process, ensuring that the temporal second-order accuracy of the proposed scheme remains uncompromised; (2). after solving the BGN2 scheme at each time step, we employ the fully-implicit BGN scheme for the trivial flow in order to achieve mesh equidistribution. Although this mesh regularization may increase the computational cost, the unaffected temporal second-order accuracy ensures that our newly proposed scheme remains more efficient than classical BGN schemes. More importantly, this mesh regularization allows for the establishment of unconditional energy stability.

The remaining of the paper is organized as follows. In Section 2, taking CSF as an example, we begin by recalling the standard BGN scheme, and then propose a second-order in time, BGN-based parametric finite element method for solving CSF. Two mesh regularization procedures are proposed to ensure the good mesh quality during the evolution. Section 3 is devoted to explaining the importance of using shape metrics, such as manifold distance and Hausdorff distance, to accurately measure the errors of two curves. We extend the proposed second-order scheme to other geometric flows such as AP-CSF and the fourth-order flow SDF in Section 4. Extensive numerical results are provided to demonstrate the accuracy and efficiency of the proposed schemes in Section 5. Finally, we draw some conclusions in Section 6.

2. For curve shortening flow (CSF)

In this section, we propose a parametric finite element method with second-order temporal accuracy for numerically solving the CSF. The same idea can be easily extended to other geometric flows (cf. Section 4). To provide a comprehensive understanding, we first review a classical first-order BGN scheme proposed by Barrett, Garcke and Nürnberg [8,9,13].

2.1. Derivation of the classical BGN scheme

To begin with, we rewrite the CSF into the following formulation as presented in Eqs. (1.2):

$$\begin{aligned} \partial_t \mathbf{X} \cdot \mathbf{n} &= -\kappa, \\ \kappa \mathbf{n} &= -\partial_{ss} \mathbf{X}. \end{aligned} \tag{2.1}$$

We introduce the following finite element approximation. Let $\mathbb{I} = [0, 1] = \bigcup_{j=1}^N I_j$, $N \geq 3$, be a decomposition of \mathbb{I} into intervals given by the nodes $\text{red } \rho_j$, $I_j = [\rho_{j-1}, \rho_j]$. Let $h = \max_{1 \leq j \leq N} |\rho_j - \rho_{j-1}|$ be the maximal length of a grid element. Define the linear finite element space as

$$V^h := \{u \in C(\mathbb{I}) : u|_{I_j} \text{ is linear, } \forall j = 1, 2, \dots, N; \quad u(\rho_0) = u(\rho_N)\} \subseteq H^1(\mathbb{I}).$$

The mass lumped inner product $(\cdot, \cdot)_{\Gamma^h}^h$ over the polygonal curve Γ^h , which is an approximation of $(\cdot, \cdot)_{\Gamma^h}$ by using the composite trapezoidal rule, is defined as

$$(u, v)_{\Gamma^h}^h := \frac{1}{2} \sum_{j=1}^N |\mathbf{X}^h(\rho_j, t) - \mathbf{X}^h(\rho_{j-1}, t)| \left[(u \cdot v)(\rho_j^-) + (u \cdot v)(\rho_{j-1}^+) \right],$$

where u, v are two scalar/vector piecewise continuous functions with possible jumps at the nodes $\{\rho_j\}_{j=1}^N$, and $u(\rho_j^\pm) = \lim_{\rho \rightarrow \rho_j^\pm} u(\rho)$.

Subsequently, the semi-discrete scheme of the formulation (2.1) is as follows: given initial polygon $\Gamma^h(0)$ with vertices lying on the initial curve $\Gamma(0)$ clockwise, parametrized by $\mathbf{X}^h(\cdot, 0) \in [V^h]^2$, find $(\mathbf{X}^h(\cdot, t), \kappa^h(\cdot, t)) \in [V^h]^2 \times V^h$ such that

$$\begin{cases} (\partial_t \mathbf{X}^h \cdot \mathbf{n}^h, \varphi^h)_{\Gamma^h}^h + (\kappa^h, \varphi^h)_{\Gamma^h}^h = 0, & \forall \varphi^h \in V^h, \\ (\kappa^h, \mathbf{n}^h \cdot \boldsymbol{\omega}^h)_{\Gamma^h}^h - (\partial_s \mathbf{X}^h, \partial_s \boldsymbol{\omega}^h)_{\Gamma^h}^h = 0, & \forall \boldsymbol{\omega}^h \in [V^h]^2, \end{cases} \tag{2.2}$$

where we always integrate over the current curve Γ^h described by \mathbf{X}^h , the outward unit normal \mathbf{n}^h is a piecewise constant vector given by

$$\mathbf{n}^h|_{I_j} = -\frac{\mathbf{h}_j^\perp}{|\mathbf{h}_j|}, \quad \mathbf{h}_j = \mathbf{X}^h(\rho_j, t) - \mathbf{X}^h(\rho_{j-1}, t), \quad j = 1, \dots, N,$$

with \cdot^\perp denoting clockwise rotation by $\frac{\pi}{2}$, and the partial derivative ∂_s is defined piecewisely over each side of the polygon $\partial_s f|_{I_j} = \frac{\partial_\rho f}{|\partial_\rho \mathbf{X}^h|}|_{I_j} = \frac{(\rho_j - \rho_{j-1}) \partial_\rho f|_{I_j}}{|\mathbf{h}_j|}$. It was shown that the scheme (2.2) will always equidistribute the vertices along Γ^h for $t > 0$ if they are not locally parallel (see Remark 2.4 in [8]).

For a full discretization, we fix $\tau > 0$ as a uniform time step size for simplicity, and let $\mathbf{X}^m \in [V^h]^2$ and Γ^m be the approximations of $\mathbf{X}(\cdot, t_m)$ and $\Gamma(t_m)$, respectively, for $m = 0, 1, 2, \dots$, where $t_m := m\tau$. We define $\mathbf{h}_j^m := \mathbf{X}^m(\rho_j) - \mathbf{X}^m(\rho_{j-1})$ and assume $|\mathbf{h}_j^m| > 0$ for $j = 1, \dots, N, \forall m > 0$. The discrete unit normal vector \mathbf{n}^m , the discrete inner product $(\cdot, \cdot)_{\Gamma^m}^h$ and the discrete operator ∂_s are defined similarly as in the semi-discrete case. Barrett, Garcke and Nürnberg used a formal first-order approximation [8,9] to replace the velocity $\partial_t \mathbf{X}$, κ and $\partial_s \mathbf{X}$ by

$$\begin{aligned} \partial_t \mathbf{X}(\cdot, t_m) &= \frac{\mathbf{X}(\cdot, t_{m+1}) - \mathbf{X}(\cdot, t_m)}{\tau} + \mathcal{O}(\tau), \\ \kappa(\cdot, t_m) &= \kappa(\cdot, t_{m+1}) + \mathcal{O}(\tau), \\ \partial_s \mathbf{X}(\cdot, t_m) &= \partial_s \mathbf{X}(\cdot, t_{m+1}) + \mathcal{O}(\tau), \end{aligned}$$

and the fully discrete semi-implicit BGN scheme (denoted as BGN1 scheme) reads as:

(BGN1, First-order in time BGN scheme for CSF): For $m \geq 0$, find $\mathbf{X}^{m+1} \in [V^h]^2$ and $\kappa^{m+1} \in V^h$ such that

$$\begin{cases} \left(\frac{\mathbf{X}^{m+1} - \mathbf{X}^m}{\tau}, \varphi^h \mathbf{n}^m \right)_{\Gamma^m}^h + (\kappa^{m+1}, \varphi^h)_{\Gamma^m}^h = 0, \quad \forall \varphi^h \in V^h, \\ (\kappa^{m+1}, \mathbf{n}^m \cdot \boldsymbol{\omega}^h)_{\Gamma^m}^h - (\partial_s \mathbf{X}^{m+1}, \partial_s \boldsymbol{\omega}^h)_{\Gamma^m} = 0, \quad \forall \boldsymbol{\omega}^h \in [V^h]^2. \end{cases} \tag{2.3}$$

The well-posedness and energy stability were established under some mild conditions. In practice, numerous numerical results show that the BGN1 scheme (2.3) converges quadratically in space [9] and linearly in time (cf. Fig. 1 in Section 5.1).

2.2. A second-order in time, BGN-based scheme

Instead of using the first-order Euler method, we apply the Crank-Nicolson leap-frog time stepping discretization in (2.2) based on the following simple calculation

$$\begin{aligned} \partial_t \mathbf{X}(\cdot, t_m) &= \frac{\mathbf{X}(\cdot, t_{m+1}) - \mathbf{X}(\cdot, t_{m-1})}{2\tau} + \mathcal{O}(\tau^2), \\ \kappa(\cdot, t_m) &= \frac{\kappa(\cdot, t_{m+1}) + \kappa(\cdot, t_{m-1})}{2} + \mathcal{O}(\tau^2), \\ \partial_s \mathbf{X}(\cdot, t_m) &= \frac{\partial_s \mathbf{X}(\cdot, t_{m+1}) + \partial_s \mathbf{X}(\cdot, t_{m-1})}{2} + \mathcal{O}(\tau^2), \end{aligned} \tag{2.4}$$

then the corresponding second-order scheme (denoted as BGN2 scheme) is as follows:

(BGN2, Second-order in time BGN-based scheme for CSF): For $\mathbf{X}^0 \in [V^h]^2, \kappa^0 \in V^h$ and $(\mathbf{X}^1, \kappa^1) \in [V^h]^2 \times V^h$ which are the appropriate approximations at the time levels $t_0 = 0$ and $t_1 = \tau$, respectively, find $\mathbf{X}^{m+1} \in [V^h]^2$ and $\kappa^{m+1} \in V^h$ for $m \geq 1$ such that

$$\begin{cases} \left(\frac{\mathbf{X}^{m+1} - \mathbf{X}^{m-1}}{2\tau}, \varphi^h \mathbf{n}^m \right)_{\Gamma^m}^h + \left(\frac{\kappa^{m+1} + \kappa^{m-1}}{2}, \varphi^h \right)_{\Gamma^m}^h = 0, \\ \left(\frac{\kappa^{m+1} + \kappa^{m-1}}{2}, \mathbf{n}^m \cdot \boldsymbol{\omega}^h \right)_{\Gamma^m}^h - \left(\frac{\partial_s \mathbf{X}^{m+1} + \partial_s \mathbf{X}^{m-1}}{2}, \partial_s \boldsymbol{\omega}^h \right)_{\Gamma^m} = 0, \end{cases} \tag{2.5}$$

for all $(\varphi^h, \boldsymbol{\omega}^h) \in V^h \times [V^h]^2$. The scheme (2.5) is semi-implicit and the computational cost is comparable to that of the BGN1 scheme (2.3). Moreover, as a temporal discretization of the semi-discrete version (2.2), it can be easily derived from (2.4) that the truncation error is of order $\mathcal{O}(\tau^2)$.

Remark 2.1. To begin the BGN2 scheme (2.5), we need to first prepare the data κ^0 and (\mathbf{X}^1, κ^1) . In practical simulations, this can be easily achieved without sacrificing the accuracy of the scheme by utilizing the standard BGN1 scheme (2.3) to get (\mathbf{X}^1, κ^1) , and the following formula of discrete curvature was proposed in [8, p. 461] to prepare κ^0 (note the sign convention of the curvature is opposite to [8])

$$\kappa^0 = (N_0^\top N_0)^{-1} N_0^\top A_0 \mathbf{X}^0, \tag{2.6}$$

where N_0 is a $2N \times N$ matrix, \mathbf{X}^0 is a $2N \times 1$ vector and A_0 is a $2N \times 2N$ matrix given by

$$\begin{aligned} N_0 &= \begin{pmatrix} (\varphi_i, (\mathbf{n}^0)^{[1]} \varphi_j)_{\Gamma^0}^h \\ (\varphi_i, (\mathbf{n}^0)^{[2]} \varphi_j)_{\Gamma^0}^h \end{pmatrix}, \quad \mathbf{X}^0 = \begin{pmatrix} \mathbf{x}^0 \\ \mathbf{y}^0 \end{pmatrix}, \\ A_0 &= \begin{pmatrix} (\partial_s \varphi_i, \partial_s \varphi_j)_{\Gamma^0} & 0 \\ 0 & (\partial_s \varphi_i, \partial_s \varphi_j)_{\Gamma^0} \end{pmatrix}, \end{aligned}$$

where $\varphi_i, 1 \leq i \leq N$ are the standard Lagrange basis over $\mathbb{1}$, and $\mathbf{a}^{[1]}, \mathbf{a}^{[2]}$ are the first and second component of vector $\mathbf{a} \in \mathbb{R}^2$, and $\mathbf{x}_j^0 = (\mathbf{X}^0)^{[1]}(\rho_j), \mathbf{y}_j^0 = (\mathbf{X}^0)^{[2]}(\rho_j)$ for $j = 1, \dots, N$. Note that this formula can be derived by solving the finite element approximation of the equation $\kappa \mathbf{n} = -\partial_{ss} \mathbf{X}$ and using the least square method. We can summarize the process as Algorithm 2.1, which outlines the

steps to prepare the required data κ^0 and (\mathbf{X}^1, κ^1) . Once we have obtained these data, we can directly apply the BGN2 scheme (2.5) to calculate (\mathbf{X}^m, κ^m) , for $m \geq 2$.

Algorithm 2.1. (Preparation for the initial data of BGN2 for CSF)

Step 0. Given the initial curve $\Gamma(0)$, the number of grid points N and the time step size τ . We choose the polygon Γ^0 with N vertices lying on $\Gamma(0)$ such that Γ^0 is (almost) equidistributed, i.e., each side of the polygon is (nearly) equal in length. We parameterize Γ^0 with $\mathbf{X}^0 \in [V^h]^2$ and the grid points ρ_j can be determined correspondingly.

Step 1. Using \mathbf{X}^0 as the input, we compute κ^0 using the discrete curvature formula (2.6).

Step 2. Using \mathbf{X}^0 as the input, we obtain (\mathbf{X}^1, κ^1) by solving the BGN1 scheme (2.3) for one time step.

Remark 2.2. When dealing with an initial curve which is not regular, an alternative approach for initialization is to solve the BGN1 scheme twice and start the BGN2 scheme from $m = 2$. Specifically, for given \mathbf{X}^0 , we can compute (\mathbf{X}^1, κ^1) and (\mathbf{X}^2, κ^2) , which are the appropriate approximations at time levels $t_1 = \tau$ and $t_2 = 2\tau$, by solving the BGN1 scheme (2.3) twice. These approximations can be used as initial values to implement the BGN2 scheme (2.3) for $m \geq 2$. For the superiority of this approach, see Fig. 6 in Section 5.3.

Similar to the BGN1 scheme (2.3), we can show the well-posedness of the BGN2 scheme (2.5) under some mild conditions as follows.

Theorem 2.1 (Well-posedness). For $m \geq 0$, we assume that the following two conditions are satisfied:

- (1) There exist at least two vectors in $\{\mathbf{h}_j^m\}_{j=1}^N$ which are not parallel, i.e.,

$$\dim \left(\text{Span} \left\{ \mathbf{h}_j^m \right\}_{j=1}^N \right) = 2.$$

- (2) No degenerate elements exist on Γ^m , i.e.,

$$\min_{1 \leq j \leq N} |\mathbf{h}_j^m| > 0.$$

Then the full discretization (2.5) is well-posed, i.e., there exists a unique solution $(\mathbf{X}^{m+1}, \kappa^{m+1}) \in [V^h]^2 \times V^h$ of (2.5).

Proof. It suffices to prove the following algebraic system for $(\mathbf{X}, \kappa) \in [V^h]^2 \times V^h$ has only zero solution,

$$\begin{cases} \left(\frac{\mathbf{X}}{\tau}, \varphi^h \mathbf{n}^m \right)_{\Gamma^m}^h + (\kappa, \varphi^h)_{\Gamma^m}^h = 0, & \forall \varphi^h \in V^h, \\ (\kappa, \mathbf{n}^m \cdot \boldsymbol{\omega}^h)_{\Gamma^m}^h - (\partial_s \mathbf{X}, \partial_s \boldsymbol{\omega}^h)_{\Gamma^m} = 0, & \forall \boldsymbol{\omega}^h \in [V^h]^2. \end{cases}$$

Indeed, the stiffness matrix is exactly the same as the standard BGN1 scheme (2.3) and thus the same argument in [9, Theorem 2.9] yields the conclusion under the assumptions (1) and (2). \square

2.3. Mesh regularization by semi-implicit BGN1 scheme

As was mentioned earlier, the semi-discrete scheme (2.2) possesses the mesh equidistribution property [13, Theorem 79]. In practice, the fully-discrete BGN1 scheme (2.3) can maintain the asymptotic long-time mesh equidistribution property. However, the BGN2 scheme (2.5) may have oscillating mesh ratio due to the structure of two-step method, which can potentially amplify the mesh ratio and cause mesh distortion or clustering during the evolution, especially for some initial curves which are not so regular, e.g., a ‘flower’ curve (see the second row of Fig. 7). Therefore, a mesh regularization procedure is necessary in real simulations to help the mesh maintain a good distribution property during the evolution, when the mesh ratio exceeds a given threshold value. Inspired by the good mesh distribution property of the BGN1 scheme, we utilize the BGN1 scheme as the mesh regularization technique. In the following, we denote n_{MR} as the threshold value chosen initially. If the mesh ratio $\Psi^m > n_{MR}$, then we use the mesh regularization procedure to improve the mesh distribution. We present a summary of the complete algorithm of BGN2 scheme for solving the CSF in Algorithm 2.2.

Algorithm 2.2. (BGN2 scheme for CSF)

Step 0. Given the initial curve $\Gamma(0)$, and N, T, n_{MR}, τ , compute \mathbf{X}^0 as in Step 0 in Algorithm 2.1.

Step 1. Using \mathbf{X}^0 as the input, we compute κ^0 using the discrete curvature formula (2.6) and solve (\mathbf{X}^1, κ^1) via the BGN1 scheme (2.3). Set $m = 1$.

Step 2. Calculate the mesh ratio Ψ^m of \mathbf{X}^m , $m \geq 1$.

Step 3. If the mesh ratio $\Psi^m > n_{MR}$, then replace (\mathbf{X}^m, κ^m) with the solution of the BGN1 scheme (2.3) with \mathbf{X}^{m-1} as the input by one run; otherwise, skip this step.

Step 4. Use the BGN2 scheme (2.5) to obtain $(\mathbf{X}^{m+1}, \kappa^{m+1})$.

Step 5. Update $m = m + 1$. If $m < T/\tau$, then go back to **Step 2**; otherwise, stop the algorithm and output the data.

As shown in **Step 3** of Algorithm 2.2, if the mesh ratio $\Psi^m > n_{MR}$, we replace (\mathbf{X}^m, κ^m) with the solution of the BGN1 scheme (2.3) with \mathbf{X}^{m-1} as the input by one run, to help us realize the mesh regularization. Extensive numerical experiments suggest that the mesh regularization procedure is very effective, and the mesh ratio decreases immediately to a small value after this procedure (cf. Fig. 4(d) in Section 5). The BGN2 scheme with the aid of the BGN1 scheme as the mesh regularization is very efficient and stable in practical simulations. The reason comes from that the BGN1 scheme (2.3) can intrinsically lead to a good mesh distribution property, which was explained in [8,13], but a more convincing explanation needs further rigorous numerical analysis for the scheme.

One concern that may arise is whether the BGN2 scheme with necessary mesh regularization can still achieve second-order accuracy, considering that the BGN1 scheme is only first-order accurate. It is important to note that for certain smooth initial curves, such as elliptic curves, the mesh regularization procedure is never required during the evolution. In such cases, the numerical evolution remains remarkably stable and the mesh ratio remains bounded. While for certain special initial curves, like a ‘flower’ curve or a ‘tube’ curve, the mesh regularization procedure may be needed only a few times (cf. Section 5.3). Nevertheless, this does not compromise the temporal second-order accuracy of the BGN2 scheme (2.5).

2.4. Mesh regularization by implicit equi-BGN1 scheme for trivial flow

In the following, we recall a fully-implicit scheme for CSF [12] which intrinsically equidistributes the vertices along the curve at each time step.

(equi-BGN1, First-order in time equidistribution BGN scheme for CSF): For $m \geq 0$, find $\mathbf{X}^{m+1} \in [V^h]^2$ and $\kappa^{m+1} \in V^h$ such that

$$\begin{cases} \left(\frac{\mathbf{X}^{m+1} - \mathbf{X}^m}{\tau}, \varphi^h \mathbf{n}^{m+1} \right)_{\Gamma^{m+1}}^h + (\kappa^{m+1}, \varphi^h)_{\Gamma^{m+1}}^h = 0, \quad \forall \varphi^h \in V^h, \\ (\kappa^{m+1}, \mathbf{n}^{m+1} \cdot \boldsymbol{\omega}^h)_{\Gamma^{m+1}}^h - (\partial_s \mathbf{X}^{m+1}, \partial_s \boldsymbol{\omega}^h)_{\Gamma^{m+1}} = 0, \quad \forall \boldsymbol{\omega}^h \in [V^h]^2. \end{cases} \tag{2.7}$$

It has been shown in [12] that

$$|\mathbf{h}_j^{m+1}| = |\mathbf{h}_{j-1}^{m+1}|, \quad \text{if } \mathbf{h}_j^{m+1} \nparallel \mathbf{h}_{j-1}^{m+1}, \quad j = 1, \dots, N. \tag{2.8}$$

Moreover, the stability estimate holds

$$L^{m+1} + \tau (\kappa^{m+1}, \kappa^{m+1})_{\Gamma^{m+1}}^h \leq L^m, \tag{2.9}$$

where L^m represents the length of the polygon Γ^m .

Inspired by the equidistribution property of the fully implicit scheme (2.7), we propose to implement the mesh regularization using the equi-BGN1 scheme for the trivial flow

$$\mathcal{V} = 0,$$

which can distribute mesh points equally and retain the shape of the curve in the continuous level. More specifically, with $\mathbf{X}^m \in [V^h]^2$, find $\tilde{\mathbf{X}}^m \in [V^h]^2$ and $\tilde{\kappa}^m \in V^h$ such that

$$\begin{cases} (\tilde{\mathbf{X}}^m - \mathbf{X}^m, \varphi^h \tilde{\mathbf{n}}^m)_{\tilde{\Gamma}^m}^h = 0, \quad \forall \varphi^h \in V^h, \\ (\tilde{\kappa}^m, \tilde{\mathbf{n}}^m \cdot \boldsymbol{\omega}^h)_{\tilde{\Gamma}^m}^h - (\partial_s \tilde{\mathbf{X}}^m, \partial_s \boldsymbol{\omega}^h)_{\tilde{\Gamma}^m} = 0, \quad \forall \boldsymbol{\omega}^h \in [V^h]^2. \end{cases} \tag{2.10}$$

Similar to (2.7), it can be rigorously proved that the vertices of $\tilde{\Gamma}^m$ are evenly distributed and the perimeter does not increase, i.e.,

$$|\tilde{\mathbf{h}}_j^m| = |\tilde{\mathbf{h}}_{j-1}^m|, \quad j = 1, \dots, N; \quad \tilde{L}^m \leq L^m. \tag{2.11}$$

Now, we present a summary of the entire algorithm for the equi-BGN2 scheme for solving the CSF. This scheme can be regarded as a variant of scheme (2.5), where $(\mathbf{X}^{m-1}, \kappa^{m-1})$ and Γ^m are replaced by their mesh regularized approximations $(\tilde{\mathbf{X}}^{m-1}, \tilde{\kappa}^{m-1})$ and $\tilde{\Gamma}^m$, respectively.

Algorithm 2.3. (equi-BGN2 scheme for CSF)

Step 0. Given the initial curve $\Gamma(0)$, and N, T, τ , compute \mathbf{X}^0 as in **Step 0** in Algorithm 2.1. Use equi-BGN1 scheme (2.10) to obtain the equidistributed polygon $\tilde{\mathbf{X}}^0$ and $\tilde{\kappa}^0$.

Step 1. Using $\tilde{\mathbf{X}}^0$ as the input, we obtain $(\tilde{\mathbf{X}}^1, \tilde{\kappa}^1)$ by solving the equi-BGN1 scheme (2.7) with time step τ . Set $m = 1$.

Step 2. Solve the BGN2 scheme (2.5) with $(\tilde{\mathbf{X}}^{m-1}, \tilde{\kappa}^{m-1})$ and $\tilde{\Gamma}^m$ to obtain $(\mathbf{X}^{m+1}, \kappa^{m+1})$.

Step 3. Update $m = m + 1$. Apply the equi-BGN1 scheme (2.10) to obtain the mesh-regularized approximation $\tilde{\mathbf{X}}^m$ and $\tilde{\kappa}^m$.

Step 4. If $m < T/\tau$, then go back to **Step 2**; otherwise, stop the algorithm and output the data $\tilde{\mathbf{X}}^m$ as an approximation solution at time t_m .

Indeed, the solution $\tilde{\mathbf{X}}^m$ not only equidistributes the vertices at each time level, but also is unconditionally stable in the following sense.

Theorem 2.2 (Interlaced energy stability). *Let $\tilde{\Gamma}^m = \tilde{\mathbf{X}}^m(\mathbb{I})$ be the solution of Algorithm 2.3. Then for any $\tau > 0$ and $m \geq 1$, the energy stability holds*

$$\tilde{L}^{m+1} \leq \tilde{L}^{m-1}, \tag{2.12}$$

where $\tilde{L}^m := \sum_{j=1}^N |\tilde{\mathbf{h}}_j^m|$ is the perimeter of $\tilde{\Gamma}^m$. In particular, we have

$$\tilde{L}^m \leq L^0, \quad \forall m \geq 1, \tag{2.13}$$

where L^0 is the perimeter of the initial polygon Γ^0 .

Proof. Recalling *Step 2*, taking $\omega^h = \frac{\mathbf{X}^{m+1} - \tilde{\mathbf{X}}^{m-1}}{2\tau}$ and $\varphi^h = \frac{\kappa^{m+1} + \tilde{\kappa}^{m-1}}{2}$ in equation (2.5), we get

$$\begin{aligned} & \left(\frac{\kappa^{m+1} + \tilde{\kappa}^{m-1}}{2}, \frac{\kappa^{m+1} + \tilde{\kappa}^{m-1}}{2} \right)_{\tilde{\Gamma}^m} h \\ &= - \left(\frac{\mathbf{X}^{m+1} - \tilde{\mathbf{X}}^{m-1}}{2\tau}, \left(\frac{\kappa^{m+1} + \tilde{\kappa}^{m-1}}{2} \right) \tilde{\mathbf{n}}^m \right)_{\tilde{\Gamma}^m} h \\ &= - \left(\frac{\partial_s \mathbf{X}^{m+1} + \partial_s \tilde{\mathbf{X}}^{m-1}}{2}, \frac{\partial_s \mathbf{X}^{m+1} - \partial_s \tilde{\mathbf{X}}^{m-1}}{2\tau} \right)_{\tilde{\Gamma}^m} h \\ &= - \frac{1}{4\tau} \left((\partial_s \mathbf{X}^{m+1}, \partial_s \mathbf{X}^{m+1})_{\tilde{\Gamma}^m} - (\partial_s \tilde{\mathbf{X}}^{m-1}, \partial_s \tilde{\mathbf{X}}^{m-1})_{\tilde{\Gamma}^m} \right). \end{aligned} \tag{2.14}$$

Noticing (2.11), we denote $|\tilde{\mathbf{h}}^m| = \frac{\tilde{L}^m}{N}$ by the length of each edge of the polygon $\tilde{\Gamma}^m$, then, we have

$$\begin{aligned} (\partial_s \mathbf{X}^{m+1}, \partial_s \mathbf{X}^{m+1})_{\tilde{\Gamma}^m} &= \sum_{j=1}^N \frac{|\mathbf{h}_j^{m+1}|}{|\tilde{\mathbf{h}}^m|} \frac{|\mathbf{h}_j^{m+1}|}{|\tilde{\mathbf{h}}^m|} |\tilde{\mathbf{h}}^m| = \sum_{j=1}^N \frac{|\mathbf{h}_j^{m+1}|^2}{|\tilde{\mathbf{h}}^m|}, \\ (\partial_s \tilde{\mathbf{X}}^{m-1}, \partial_s \tilde{\mathbf{X}}^{m-1})_{\tilde{\Gamma}^m} &= \sum_{j=1}^N \frac{|\tilde{\mathbf{h}}_j^{m-1}|}{|\tilde{\mathbf{h}}^m|} \frac{|\tilde{\mathbf{h}}_j^{m-1}|}{|\tilde{\mathbf{h}}^m|} |\tilde{\mathbf{h}}^m| = \sum_{j=1}^N \frac{|\tilde{\mathbf{h}}_j^{m-1}|^2}{|\tilde{\mathbf{h}}^m|}. \end{aligned}$$

Therefore, by combining with the Cauchy-Schwarz inequality, we can estimate

$$\begin{aligned} & (\partial_s \mathbf{X}^{m+1}, \partial_s \mathbf{X}^{m+1})_{\tilde{\Gamma}^m} - (\partial_s \tilde{\mathbf{X}}^{m-1}, \partial_s \tilde{\mathbf{X}}^{m-1})_{\tilde{\Gamma}^m} \\ &= \sum_{j=1}^N \frac{|\mathbf{h}_j^{m+1}|^2}{|\tilde{\mathbf{h}}^m|} - \sum_{j=1}^N \frac{|\tilde{\mathbf{h}}_j^{m-1}|^2}{|\tilde{\mathbf{h}}^m|} = \frac{N}{\tilde{L}^m} \sum_{j=1}^N |\mathbf{h}_j^{m+1}|^2 - \frac{(\tilde{L}^{m-1})^2}{\tilde{L}^m} \\ &\geq \frac{N}{\tilde{L}^m} \left(\sum_{j=1}^N |\mathbf{h}_j^{m+1}| \right)^2 / N - \frac{(\tilde{L}^{m-1})^2}{\tilde{L}^m} = \frac{(L^{m+1})^2 - (\tilde{L}^{m-1})^2}{\tilde{L}^m} \\ &\geq \frac{(\tilde{L}^{m+1})^2 - (\tilde{L}^{m-1})^2}{\tilde{L}^m}, \end{aligned} \tag{2.15}$$

where for the last inequality we used (2.11). Combining (2.14) and (2.15), we can deduce (2.12).

To show (2.13), it suffices to prove $\tilde{L}^0 \leq L^0$ and $\tilde{L}^1 \leq L^0$. This can be easily obtained by recalling *Step 0*, *Step 1*, (2.11) and (2.9). \square

Remark 2.3. It is also feasible to perform the mesh regularization using the semi-implicit BGN1 scheme (2.3) for the trivial flow in *Step 3* of Algorithm 2.3 at each time step. While this approach can reduce the global computational costs, achieving a theoretical proof of energy stability, as demonstrated in Theorem 2.2, seems unattainable.

In subsequent sections, we will denote (2.3) and (2.7) by the BGN1 and equi-BGN1 scheme, respectively. We call Algorithm 2.2 and Algorithm 2.3 as the BGN2 and equi-BGN2 scheme, respectively.

3. Shape metric is a better choice

As we are aware, it is an interesting and thought-provoking problem to determine how to quantify the difference between two curves in 2D or two surfaces in 3D. Given two closed curves Γ_1 and Γ_2 , we assume that the two curves are parametrized by $\mathbf{X}(\rho)$ and $\mathbf{Y}(\rho)$, respectively, over the same interval I . Consequently, we can define the following four metrics for measurement:

- (L^2 -error) The L^2 -norm between the parametrized functions $\mathbf{X}(\rho)$ and $\mathbf{Y}(\rho)$ is defined in a classical way

$$A(\mathbf{X}, \mathbf{Y}) := \|\mathbf{X}(\rho) - \mathbf{Y}(\rho)\|_{L^2(I)}.$$

- (L^∞ -error) The L^∞ -norm between the parametrized functions $\mathbf{X}(\rho)$ and $\mathbf{Y}(\rho)$ is defined as

$$B(\mathbf{X}, \mathbf{Y}) := \|\mathbf{X}(\rho) - \mathbf{Y}(\rho)\|_{L^\infty(I)}.$$

- (**Manifold distance**) The manifold distance between the curves Γ_1 and Γ_2 is defined as [31]

$$M(\Gamma_1, \Gamma_2) := |(\Omega_1 \setminus \Omega_2) \cup (\Omega_2 \setminus \Omega_1)| = |\Omega_1| + |\Omega_2| - 2|\Omega_1 \cap \Omega_2|,$$

where Ω_1 and Ω_2 represent the regions enclosed by Γ_1 and Γ_2 , respectively, and $|\Omega|$ denotes the area of Ω .

- (**Hausdorff distance**) The Hausdorff distance between the curves Γ_1 and Γ_2 is defined as [2]

$$H(\Gamma_1, \Gamma_2) = \max\{\tilde{H}(\Gamma_1, \Gamma_2), \tilde{H}(\Gamma_2, \Gamma_1)\},$$

where $\tilde{H}(\Gamma_1, \Gamma_2) = \max_{a \in \Gamma_1} \min_{b \in \Gamma_2} d(a, b)$, and d is the Euclidean distance.

Remark 3.1. The L^2 -error and L^∞ -error fall within the domain of *function metrics*, which rely on the parametrization of curves. On the other hand, as demonstrated in [31, Proposition 5.1] and [2], it has been easily proven that both manifold distance and Hausdorff distance fulfill the properties of symmetry, positivity and the triangle inequality. Therefore, they belong to the category of *shape metrics* and not influenced by the specific parametrization.

Remark 3.2. It should be noted that the aforementioned shape metrics can be easily calculated using simple algorithms. As the numerical solutions are represented as polygons, it is very easy to calculate the area of the symmetric difference region, i.e., the manifold distance, between two polygonal curves. Additionally, a polygon-based approach proposed in the literature [2] can be employed to calculate the Hausdorff distance between planar curves.

In order to test the convergence rate of numerical schemes, for example, we consider the evolution of the CSF with an initial ellipse defined by

$$\{(x, y) \in \mathbb{R}^2 : x^2 + 4y^2 = 4\}.$$

This initial ellipse is approximated using an equidistributed polygon \mathbf{X}^0 with N vertices. Here, we simulate the CSF by using three different numerical schemes: Dziuk's scheme [17, Section 6], BGN1 scheme and BGN2 scheme. Since the exact solution of the CSF for an elliptical curve is unknown, we first compute a reference solution \mathbf{X}_{ref} by Dziuk's scheme (to test the convergence of Dziuk's scheme) or the BGN2 scheme (to test the convergence of BGN-type schemes) with a fine mesh and a tiny time step size, e.g., $N = 10000$ and $\tau = 10^{-1} * 2^{-11}$. To test the temporal error, we still take a large number of grid points, e.g., $N = 10000$, such that the spatial error is ignorable. The numerical error and the corresponding convergence order are then determined as follows

$$\mathcal{E}_{\mathcal{M}} := \mathcal{E}_\tau(T) = \mathcal{M}(\mathbf{X}_\tau^k, \mathbf{X}_{\text{ref}}), \quad \text{Order} = \log\left(\frac{\mathcal{E}_\tau(T)}{\mathcal{E}_{\tau/2}(T)}\right) / \log 2, \tag{3.1}$$

where $k = T/\tau$, and \mathcal{M} represents any one of the four metrics defined above.

Tables 1-3 display the numerical errors at time $T = 0.25$ measured by the four different metrics for Dziuk's scheme [17], the BGN1 scheme and the BGN2 scheme, respectively. As anticipated, we easily observe linear convergence in time for Dziuk's scheme across all four different metrics. While linear and quadratic convergence for both shape metrics (i.e., the manifold distance and Hausdorff distance) are observed for the BGN1 scheme in Table 2 and the BGN2 scheme in Table 3, respectively.

It is worth noting that unlike Dziuk's scheme, the convergence of the BGN1 scheme and BGN2 scheme under function metrics (the L^2 -norm and L^∞ -norm) is not as satisfactory. This is not surprising since the error in classical Sobolev space depends on the specific parametrization of the curve. In contrast, the BGN formulation (2.1) allows tangential motion to make the mesh points equidistribute, which indeed affects the parametrization while preserving the shape of the curve. Thus it is not appropriate to use the classical function metrics to quantify the errors of the BGN-type schemes which are based on the BGN formulation. Instead, as observed from Tables 2 and 3, the shape metrics are much more suitable for quantifying the numerical errors of the schemes that allow intrinsic tangential velocity. In the remaining of the article, we will employ the manifold distance or the Hausdorff distance when measuring the difference between two curves.

Table 1
Numerical errors quantified by various metrics for Dziuk’s scheme [17, Section 6], with the parameters $N = 10000$, $\tau_0 = 1/40$, and $T = 0.25$.

Errors	$\tau = \tau_0$	$\tau_0/2$	$\tau_0/2^2$	$\tau_0/2^3$
L^2 -norm	1.17E-2	6.31E-3	3.26E-3	1.62E-3
Order	–	0.89	0.95	1.01
L^∞ -norm	3.05E-2	1.63E-2	8.41E-3	4.19E-3
Order	–	0.90	0.96	1.00
Manifold distance	6.89E-2	3.65E-2	1.86E-2	9.17E-3
Order	–	0.92	0.97	1.02
Hausdorff distance	3.04E-2	1.62E-2	8.29E-3	4.09E-3
Order	–	0.91	0.97	1.02

Table 2
Numerical errors quantified by various metrics for the BGN1 scheme, with the parameters $N = 10000$, $\tau_0 = 1/40$, $T = 0.25$.

Errors	$\tau = \tau_0$	$\tau_0/2$	$\tau_0/2^2$	$\tau_0/2^3$
L^2 -norm	4.25E-3	3.98E-3	4.05E-3	4.15E-3
Order	–	0.10	–0.03	–0.03
L^∞ -norm	1.00E-2	9.17E-3	9.47E-3	9.79E-3
Order	–	0.12	–0.05	–0.05
Manifold distance	3.11E-2	1.58E-2	7.96E-3	4.00E-3
Order	–	0.98	0.99	0.99
Hausdorff distance	8.23E-3	4.18E-3	2.11E-3	1.06E-3
Order	–	0.98	0.99	0.99

Table 3
Numerical errors quantified by various metrics for the BGN2 scheme, with the parameters $N = 10000$, $\tau_0 = 1/40$, $T = 0.25$.

Errors	$\tau = \tau_0$	$\tau_0/2$	$\tau_0/2^2$	$\tau_0/2^3$
L^2 -norm	1.49E-2	1.45E-2	1.45E-2	1.43E-2
Order	–	0.04	0.00	0.02
L^∞ -norm	3.32E-2	3.30E-2	3.29E-2	3.29E-2
Order	–	0.01	0.00	0.00
Manifold distance	8.44E-4	2.11E-4	5.27E-5	1.32E-5
Order	–	2.00	2.00	1.99
Hausdorff distance	2.00E-4	4.98E-5	1.26E-5	3.29E-6
Order	–	2.01	1.98	1.94

4. Applications to other geometric flows

In this section, we extend the above proposed BGN2 scheme to other geometric flows.

4.1. For area-preserving curve-shortening flow (AP-CSF)

As is known, the AP-CSF can be viewed as the L^2 -gradient flow with respect to the length functional under the constraint of total area preservation [13,23]. Similar to (2.1), we rewrite the AP-CSF as the following coupled equations

$$\begin{aligned} \partial_t \mathbf{X} \cdot \mathbf{n} &= -\kappa + \langle \kappa \rangle, \\ \kappa \mathbf{n} &= -\partial_{ss} \mathbf{X}, \end{aligned} \tag{4.1}$$

where the average of curvature is defined as $\langle \kappa \rangle := \int_{\Gamma(t)} \kappa ds / \int_{\Gamma(t)} 1 ds$. The fully-discrete, first-order in time semi-implicit BGN scheme for AP-CSF reads as [13]:

(BGN1 scheme for AP-CSF): For $m \geq 0$, find $\mathbf{X}^{m+1} \in [V^h]^2$ and $\kappa^{m+1} \in V^h$ such that

$$\begin{cases} \left(\frac{\mathbf{X}^{m+1} - \mathbf{X}^m}{\tau}, \varphi^h \mathbf{n}^m \right)_{\Gamma^m}^h + (\kappa^{m+1} - \langle \kappa^{m+1} \rangle_{\Gamma^m}, \varphi^h)_{\Gamma^m}^h = 0, \\ (\kappa^{m+1}, \mathbf{n}^m \cdot \boldsymbol{\omega}^h)_{\Gamma^m}^h - (\partial_s \mathbf{X}^{m+1}, \partial_s \boldsymbol{\omega}^h)_{\Gamma^m} = 0, \end{cases} \quad (4.2)$$

for all $(\varphi^h, \boldsymbol{\omega}^h) \in V^h \times [V^h]^2$, where $\langle \kappa^{m+1} \rangle_{\Gamma^m} := (\kappa^{m+1}, 1)_{\Gamma^m}^h / (1, 1)_{\Gamma^m}^h$.

Based on the same spirit, we can propose the following second-order BGN2 scheme.

(BGN2 scheme for AP-CSF): For $m \geq 1$, find $(\mathbf{X}^{m+1}, \kappa^{m+1}) \in [V^h]^2 \times V^h$ such that

$$\begin{cases} \left(\frac{\mathbf{X}^{m+1} - \mathbf{X}^{m-1}}{2\tau}, \varphi^h \mathbf{n}^m \right)_{\Gamma^m}^h = - \left(\frac{\kappa^{m+1} + \kappa^{m-1}}{2} - \langle \frac{\kappa^{m+1} + \kappa^{m-1}}{2} \rangle_{\Gamma^m}, \varphi^h \right)_{\Gamma^m}^h, \\ \left(\frac{\kappa^{m+1} + \kappa^{m-1}}{2}, \mathbf{n}^m \cdot \boldsymbol{\omega}^h \right)_{\Gamma^m}^h - \left(\frac{\partial_s \mathbf{X}^{m+1} + \partial_s \mathbf{X}^{m-1}}{2}, \partial_s \boldsymbol{\omega}^h \right)_{\Gamma^m} = 0, \end{cases} \quad (4.3)$$

for all $(\varphi^h, \boldsymbol{\omega}^h) \in V^h \times [V^h]^2$. Similarly, the stiffness matrix of the linear system to be solved in (4.3) is exactly the same as the BGN1 scheme (4.2), whose well-posedness has been established in [13, Theorem 90]. The equi-BGN1 scheme [13] and equi-BGN2 scheme can be derived in a similar manner. Similarly, unconditional interlaced energy stability for the equi-BGN2 scheme can be obtained.

4.2. For surface diffusion flow (SDF)

We consider the fourth-order flow—SDF, which can be viewed as the H^{-1} -gradient flow with respect to the length functional [7, 13]. In a similar fashion, we rephrase the SDF as the subsequent system of equations

$$\begin{aligned} \partial_t \mathbf{X} \cdot \mathbf{n} &= \partial_{ss} \kappa, \\ \kappa \mathbf{n} &= -\partial_{ss} \mathbf{X}. \end{aligned} \quad (4.4)$$

The fully discrete, first-order in time semi-implicit BGN scheme for SDF reads as [8]:

(BGN1 scheme for SDF): For $m \geq 0$, find $\mathbf{X}^{m+1} \in [V^h]^2$ and $\kappa^{m+1} \in V^h$ such that

$$\begin{cases} \left(\frac{\mathbf{X}^{m+1} - \mathbf{X}^m}{\tau}, \varphi^h \mathbf{n}^m \right)_{\Gamma^m}^h + (\partial_s \kappa^{m+1}, \partial_s \varphi^h)_{\Gamma^m} = 0, \quad \forall \varphi^h \in V^h, \\ (\kappa^{m+1}, \mathbf{n}^m \cdot \boldsymbol{\omega}^h)_{\Gamma^m}^h - (\partial_s \mathbf{X}^{m+1}, \partial_s \boldsymbol{\omega}^h)_{\Gamma^m} = 0, \quad \forall \boldsymbol{\omega}^h \in [V^h]^2. \end{cases} \quad (4.5)$$

In line with the same approach, we can put forward the subsequent second-order BGN2 scheme:

(BGN2 scheme for SDF): For $m \geq 1$, find $(\mathbf{X}^{m+1}, \kappa^{m+1}) \in [V^h]^2 \times V^h$ such that

$$\begin{cases} \left(\frac{\mathbf{X}^{m+1} - \mathbf{X}^{m-1}}{2\tau}, \varphi^h \mathbf{n}^m \right)_{\Gamma^m}^h + \left(\frac{\partial_s \kappa^{m+1} + \partial_s \kappa^{m-1}}{2}, \partial_s \varphi^h \right)_{\Gamma^m} = 0, \\ \left(\frac{\kappa^{m+1} + \kappa^{m-1}}{2}, \mathbf{n}^m \cdot \boldsymbol{\omega}^h \right)_{\Gamma^m}^h - \left(\frac{\partial_s \mathbf{X}^{m+1} + \partial_s \mathbf{X}^{m-1}}{2}, \partial_s \boldsymbol{\omega}^h \right)_{\Gamma^m} = 0, \end{cases} \quad (4.6)$$

for all $(\varphi^h, \boldsymbol{\omega}^h) \in V^h \times [V^h]^2$. The well-posedness of the above scheme can be shown similarly under certain mild conditions.

For the schemes (4.3) and (4.6), we consistently set $\mathbf{X}^0 \in [V^h]^2$ as specified in Algorithm 2.1, that is, \mathbf{X}^0 is a parametrization of an (almost) equidistributed interpolation polygon with N vertices for the initial curve $\Gamma(0)$. Similar as the case of CSF, to start the BGN2 schemes, we need to prepare the initial data κ^0 and (\mathbf{X}^1, κ^1) , which can be achieved by using the similar approach as Algorithm 2.1 by using the corresponding BGN1 scheme. A complete second-order scheme can be obtained as in Algorithm 2.2 with the corresponding BGN1 scheme as a mesh regularization when necessary.

5. Numerical results

5.1. Convergence tests

In this subsection, we test the temporal convergence of the second-order schemes (2.5), (4.3) and (4.6) for solving the three geometric flows: CSF, AP-CSF and SDF, respectively, with two different mesh regularization techniques. As previously discussed in Section 3, we quantify the numerical errors of the curves using the shape metrics, such as the manifold distance and Hausdorff distance. For the following simulations, we select four distinct types of initial shapes:

- **(Shape 1):** a unit circle;
- **(Shape 2):** an ellipse with semi-major axis 2 and semi-minor axis 1;
- **(Shape 3):** a ‘tube’ shape, which is a curve comprising a 4×1 rectangle with two semicircles on its left and right sides;
- **(Shape 4):** a ‘flower’ shape, which is parameterized by

$$\mathbf{X}(\rho) = ((2 + \cos(12\pi\rho)) \cos(2\pi\rho), (2 + \cos(12\pi\rho)) \sin(2\pi\rho)), \quad \rho \in \mathbb{I} = [0, 1].$$

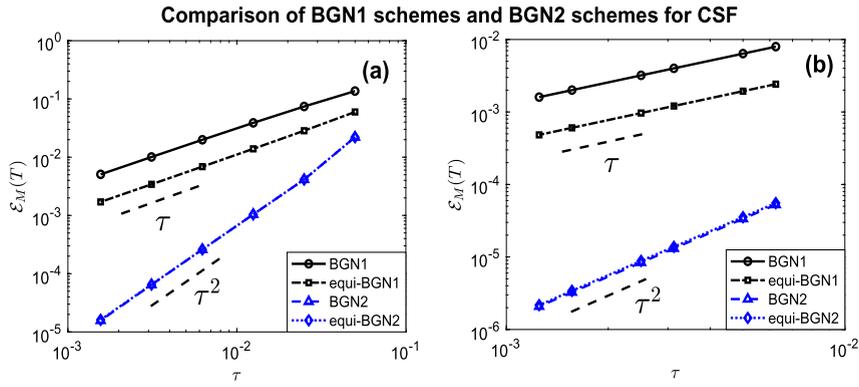


Fig. 1. Log-log plot of the numerical errors at time $T = 0.25$ measured by the manifold distance for BGN1, equi-BGN1, BGN2 and equi-BGN2 schemes for solving the CSF with two different initial curves: (a) Shape 1 and (b) Shape 2, respectively, where the number of nodes is fixed as $N = 10000$.

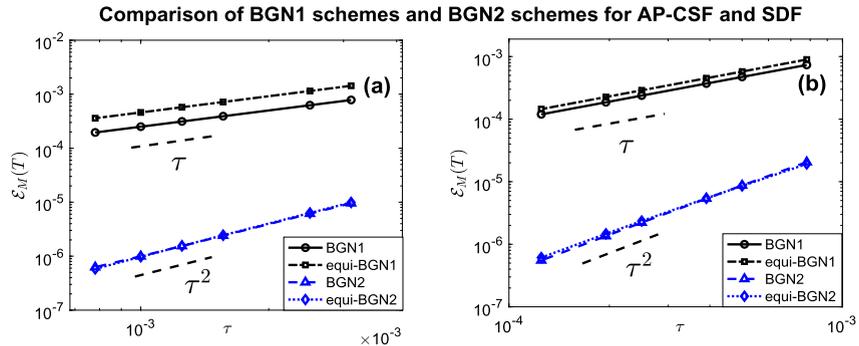


Fig. 2. Log-log plot of the numerical errors at time $T = 0.25$, measured by the manifold distance, for solving two different flows with Shape 2 as the initial curve: (a) AP-CSF and (b) SDF, respectively.

We note that for the CSF with Shape 1 as its initial shape has the following true solution, i.e.,

$$\mathbf{x}_{\text{true}}(\rho, t) = \sqrt{1 - 2t}(\cos(2\pi\rho), \sin(2\pi\rho)), \quad \rho \in \mathbb{I}, \quad t \in [0, 0.5).$$

For this particular case, we compute the numerical error by comparing it with the true solution. However, for all other cases, we utilize the reference solutions which are obtained by the BGN2 scheme with large N and a tiny time step size τ . In addition, the mesh regularization threshold for the BGN2 scheme is consistently set to $n_{\text{MR}} = 10$, and the iteration tolerance of the equi-BGN2 scheme is set as $\text{tol} = 10^{-10}$.

We begin our test by calculating the convergence of the BGN2 scheme and the equi-BGN2 scheme for the CSF with either Shape 1 or Shape 2 as initial data. Fig. 1 presents a log-log plot of the numerical errors at time $T = 0.25$, measured by the manifold distance. The errors for the Hausdorff distance, which are similar, are not included here for brevity. To ensure a fair comparison, we also include the numerical results of the BGN1 scheme (2.3) and the equi-BGN1 scheme (2.7) under the same computational parameters, with a fixed number of grid points $N = 10000$. As clearly shown in Fig. 1, the numerical error of the BGN2-type schemes reduces very rapidly with second-order accuracy in time, while the BGN1-type schemes only achieve first-order convergence.

Fig. 2 illustrates the temporal errors of the schemes for solving the AP-CSF and SDF with Shape 2 as initial data, showing quadratic convergence for BGN2-type schemes and linear convergence for BGN1-type schemes.

5.2. Comparison of computational costs

In order to show that the computational cost of the proposed BGN2 schemes is comparable to that of the BGN1 schemes, we present two examples about solving the CSF and SDF, respectively. The numerical codes were written by using MATLAB 2021b, and they were implemented in a MacBook Pro with 1.4 GHz quad-core Intel Core i5 and 8 GB RAM.

Table 4 displays a comparison of CPU times in seconds and numerical errors at time $T = 0.05$, as measured by the manifold distance $\mathcal{E}_M(T)$ and Hausdorff distance $\mathcal{E}_H(T)$, using the BGN1-type and BGN2-type schemes for solving the CSF, where the initial shape is chosen as Shape 1. Table 5 provides similar results for solving the SDF with Shape 3 as its initial shape. Based on the findings presented in Tables 4 and 5, the following conclusions can be drawn. (i) On the same mesh, the computational cost of the BGN2 scheme is slightly higher than that of the BGN1 scheme, as it involves additional calculations for the initial values and the right-hand side of the linear system at each time level. Meanwhile, the equi-BGN2 scheme incurs more or less similar computational cost as the equi-BGN1 scheme. However, the numerical solutions obtained using the BGN2-type schemes are significantly more accurate than

Table 4

Comparisons of the CPU times (seconds) and the numerical errors measured from the manifold distance $\mathcal{E}_M(T)$ and Hausdorff distance $\mathcal{E}_H(T)$ for the BGN1-type and BGN2-type schemes applied to CSF, where the initial shape is chosen as Shape 1, with $\tau = 0.5/N$ and $T = 0.05$.

BGN1 scheme				BGN2 scheme			
N	$\mathcal{E}_M(T)$	$\mathcal{E}_H(T)$	Time(s)	N	$\mathcal{E}_M(T)$	$\mathcal{E}_H(T)$	Time(s)
320	5.61E-4	1.25E-4	0.350	320	2.09E-4	5.04E-5	0.430
640	3.34E-4	6.37E-5	1.70	640	5.20E-5	1.27E-5	2.30
1280	1.81E-4	3.22E-5	9.85	1280	1.29E-5	3.20E-6	12.9
2560	9.38E-5	1.62E-5	110	2560	3.08E-6	8.16E-7	130
equi-BGN1 scheme				equi-BGN2 scheme			
N	$\mathcal{E}_M(T)$	$\mathcal{E}_H(T)$	Time(s)	N	$\mathcal{E}_M(T)$	$\mathcal{E}_H(T)$	Time(s)
320	4.70E-4	9.42E-5	1.16	320	2.09E-4	5.03E-5	0.82
640	1.82E-4	3.44E-5	4.93	640	5.20E-5	1.26E-5	4.19
1280	7.78E-5	1.40E-5	25.5	1280	1.29E-5	3.14E-6	25.4
2560	3.55E-5	6.22E-6	284	2560	3.08E-6	7.86E-7	304

Table 5

Comparisons of the CPU times (seconds) and the numerical errors measured by the manifold distance $\mathcal{E}_M(T)$ and Hausdorff distance $\mathcal{E}_H(T)$ using the BGN1-type and BGN2-type schemes applied to SDF, where the initial shape is chosen as Shape 3, with $\tau = 0.5/N$, and $T = 0.05$.

BGN1 scheme				BGN2 scheme			
N	$\mathcal{E}_M(T)$	$\mathcal{E}_H(T)$	Time(s)	N	$\mathcal{E}_M(T)$	$\mathcal{E}_H(T)$	Time(s)
320	4.73E-3	6.91E-4	0.470	320	2.53E-3	1.14E-3	0.610
640	2.24E-3	3.38E-4	2.03	640	8.28E-4	4.17E-4	2.27
1280	1.10E-3	1.67E-4	12.6	1280	2.30E-4	1.12E-4	15.1
2560	5.53E-4	8.34E-5	133	2560	5.42E-5	2.82E-5	153
equi-BGN1 scheme				equi-BGN2 scheme			
N	$\mathcal{E}_M(T)$	$\mathcal{E}_H(T)$	Time(s)	N	$\mathcal{E}_M(T)$	$\mathcal{E}_H(T)$	Time(s)
320	5.00E-3	1.04E-3	3.39	320	2.71E-3	1.21E-3	3.48
640	2.62E-3	5.61E-4	17.1	640	8.88E-4	4.33E-4	16.7
1280	1.34E-3	2.93E-4	105	1280	2.64E-4	1.56E-4	102
2560	6.83E-4	1.51E-4	1151	2560	8.12E-5	5.57E-5	1140

those of the BGN1-type schemes; (ii) The computational cost of the equi-BGN2 scheme is several times higher than that of the BGN2 scheme, since it needs to solve a nonlinear system at each time step. However, equidistribution and unconditional energy stability can be theoretically guaranteed for the equi-BGN2 scheme.

5.3. Applications to the curve evolution

As is well-known, the AP-CSF and SDF possess some structure-preserving properties, such as the perimeter decreasing and area conserving properties [7,22,23]. In this subsection, we investigate the structure-preserving properties of the proposed BGN2 scheme and equi-BGN2 scheme applied to AP-CSF and SDF, respectively. As an example, we mainly focus on the SDF here. Moreover, we will discuss the importance of the mesh regularization procedures.

Fig. 3 (a) illustrates the evolution of an initially elliptic curve, referred to as Shape 2, driven by SDF towards its equilibrium state by the BGN2 scheme. Fig. 3(b)-(e) show the evolution of various geometric quantities during the process: the relative area loss $\Delta A(t)$, the normalized perimeter $L(t)/L(0)$, and the mesh distribution function $\Psi(t)$, which are defined respectively as

$$\Delta A(t)|_{t=t_m} = \frac{A^m - A^0}{A^0}, \quad \frac{L(t)}{L(0)}|_{t=t_m} = \frac{L^m}{L^0}, \quad \Psi(t)|_{t=t_m} = \frac{\max_j |\mathbf{h}_j^m|}{\min_j |\mathbf{h}_j^m|},$$

for $m \geq 0$, where A^m is the area enclosed by the polygon determined by \mathbf{X}^m , L^m represents the perimeter of the polygon, and $\Psi(t)$ is the mesh ratio function. As depicted in Fig. 3(b), the area loss exhibits a weakly oscillating behavior, which may result from the two-step structure of the BGN2 scheme, the equi-BGN2 scheme has similar oscillating behavior and we omit it here for brevity. It is worth noting that despite the oscillations, the normalized area loss remains very low, consistently below 0.01%. By employing a smaller grid size, the area loss can be further reduced, and it is significantly lower than that of the BGN1 scheme under the same computational parameters. Furthermore, Fig. 3(c) shows the BGN2 scheme and the equi-BGN2 scheme preserve the perimeter-decreasing property of the SDF numerically. Furthermore, in Fig. 3(d), it can be observed that the mesh distribution function $\Psi(t)$ remains lower than 1.2 during the evolution. This indicates that the mesh distribution remains well-maintained and almost equidistributed during the

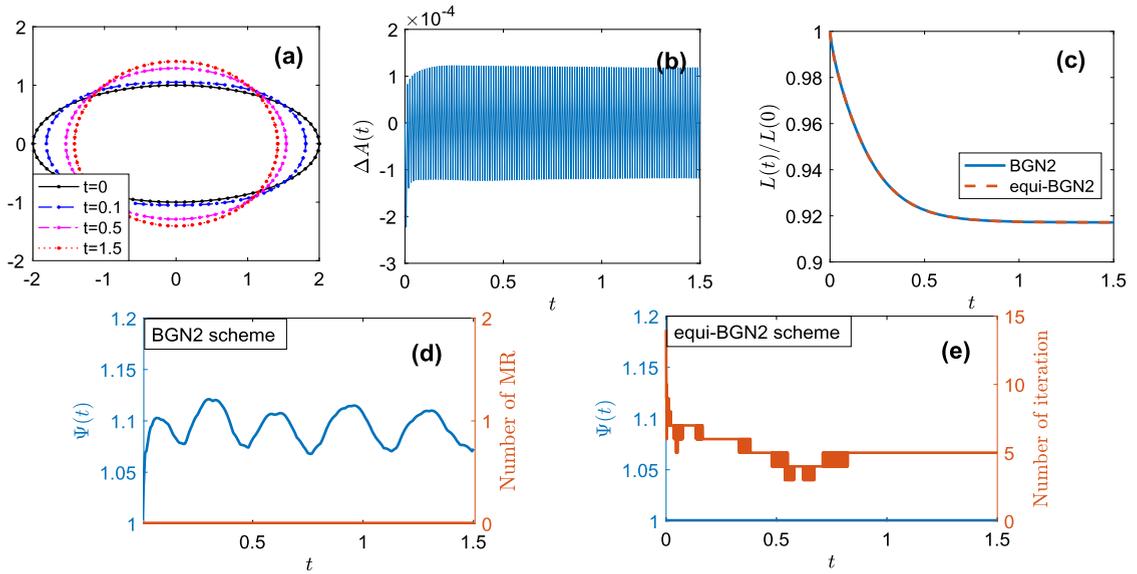


Fig. 3. (a) Several snapshots of the curve evolution controlled by the SDF, starting with Shape 2 as its initial shape. (b) The relative area loss as a function of time. (c) The normalized perimeter as a function of time. (d) The mesh ratio function $\Psi(t)$ (in blue line) and the number of mesh regularizations (in red line) for the BGN2 scheme. (e) The mesh ratio function $\Psi(t)$ (in blue line) and the number of iteration numbers (in red line) at each time step for the equi-BGN2 scheme. For (a)-(b), we used $N = 80$ and $\tau = 1/160$ while for (c)-(e), $N = 640$ and $\tau = 1/1280$. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

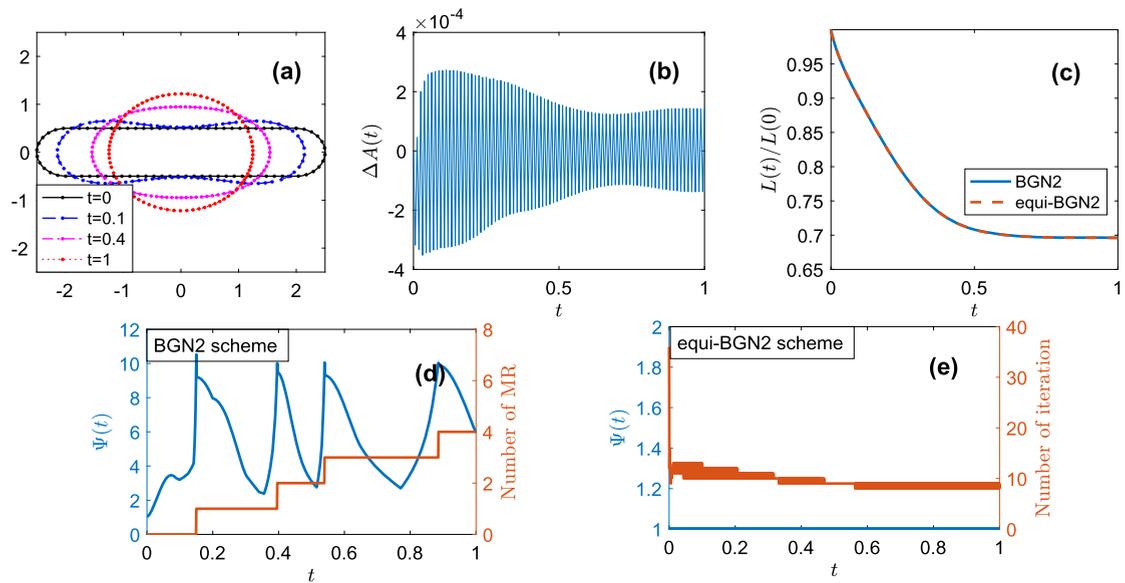


Fig. 4. (a) Several snapshots of the curve evolution controlled by the SDF, starting with Shape 3 as its initial shape. (b) The relative area loss as a function of time. (c) The normalized perimeter as a function of time. (d) The mesh distribution function $\Psi(t)$ (in blue line) and the number of mesh regularizations (in red line) for the BGN2 scheme. (e) The mesh ratio function $\Psi(t)$ (in blue line) and the number of iteration numbers (in red line) at each time step for the equi-BGN2 scheme. For (a)-(b) we used $N = 80$ and $\tau = 1/160$ while $N = 640$ and $\tau = 1/1280$ for (c)-(e).

process. Therefore, in this scenario, there is no need to perform the mesh regularization procedure because $\Psi(t)$ is always smaller than the chosen threshold n_{MR} (here we choose it as 10) in the simulations. Additionally, Fig. 3(e) shows the equi-BGN2 scheme achieves equidistribution property at each time step. The relatively low iteration numbers do not compromise the computational efficiency.

To provide a more comprehensive comparison, we conduct simulations of evolution of Shape 3 curve driven by the SDF. Fig. 4(b)-(c) demonstrates that the BGN2 scheme and the equi-BGN2 scheme effectively preserve two crucial geometric properties of the SDF: the conservation of area and the reduction of perimeter properties [7,22]. It should be noted that Fig. 4(d) reveals that without the implementation of mesh regularization, the mesh distribution function $\Psi(t)$ can become very large. Therefore, in our algorithm,

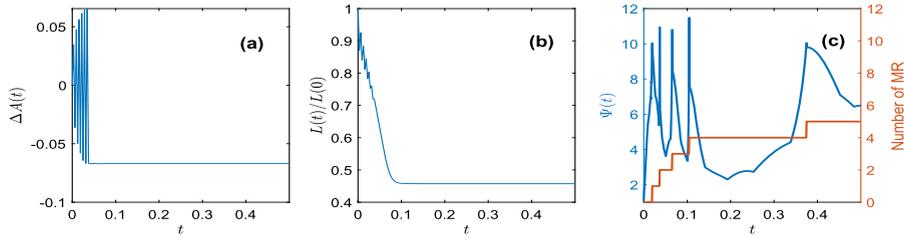


Fig. 5. Evolution of the three geometrical quantities when the initial data is prepared as in Algorithm 2.1: (a) the relative area loss, (b) the normalized perimeter, (c) the mesh distribution function $\Psi(t)$, for the BGN2 scheme.

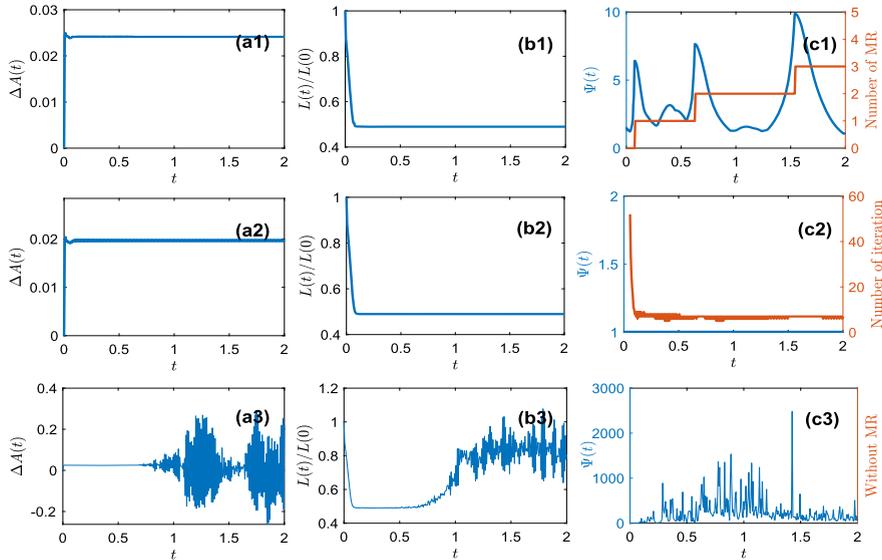


Fig. 6. Evolution of the three geometrical quantities when the initial data is prepared as in Remark 2.2: (a) the relative area loss, (b) the normalized perimeter, (c) the mesh distribution function $\Psi(t)$, for the BGN2 scheme (shown in the first row), the equi-BGN2 scheme (shown in the second row) and without mesh regularization procedure (shown in the third row).

when $\Psi(t)$ exceeds a threshold n_{MR} , we employ the BGN1 scheme (4.5) for a single run to perform mesh regularization, similar to Step 3 of Algorithm 2.2. As clearly shown in Fig. 4(d), following this step, the mesh ratio rapidly decreases to a low value, which makes the method more stable. Importantly, this mesh regularization procedure is only required four times throughout the entire evolution, without sacrificing the accuracy of the BGN2 scheme (cf. Table 5). Similarly, as shown in Fig. 4(e), the equi-BGN2 scheme also performs well for this initial shape. Compared to the case of Shape 2, although we require more iteration steps, it is still superior to the BGN1 scheme in view of the accuracy and efficiency (cf. Table 5).

Next, we proceed to simulate the evolution of a nonconvex curve, referred to as Shape 4. Fig. 5 and Fig. 6 (the first row) show the evolution of the geometric quantities based on two different initial data preparations: Algorithm 2.1 and Remark 2.2, respectively. A comparison of the results reveals the superiority of the latter approach for several reasons: (i) the magnitude of area loss is significantly lower when using the approach in Remark 2.2; (ii) the perimeter-decreasing property is preserved while the perimeter oscillates at the beginning when using Algorithm 2.1; (iii) the number of mesh regularization implementations is smaller with the approach in Remark 2.2. Thus we recommend preparing the data for a nonconvex initial curve following the approach outlined in Remark 2.2. Fig. 6 (the second row) demonstrates the performance of the equi-BGN2 scheme, from which it can be seen that only a relatively low number of iterations are needed for the majority of time steps (see 6(c2)). Additionally, Fig. 6 (the third row) illustrates the evolution of the same quantities without any implementations of mesh regularization. In this case, all three quantities exhibit significant oscillations after a certain time period, and the area loss and mesh ratio of the polygon becomes excessively large, resulting in the breakdown of the BGN2 scheme. Notably, mesh clustering has happened at $t = 1$ (see Fig. 7(c3)), eventually leading to mesh distortion at $t = 2$ (see Fig. 7(d3)).

These issues can be avoided by implementing one of the mesh regularization techniques. Fig. 7(a1)-(d1) and Fig. 7(a2)-(d2) demonstrate that mesh regularization is crucial for the effectiveness of BGN2-type schemes and the BGN1-type schemes can significantly enhance mesh quality. Additionally, A comparison between Fig. 7(d1) and Fig. 7(d2) reveals that there still exists some mesh clustering for the BGN2 scheme in long-time evolution. In contrast, the equi-BGN2 scheme exhibits equidistribution property throughout all time.

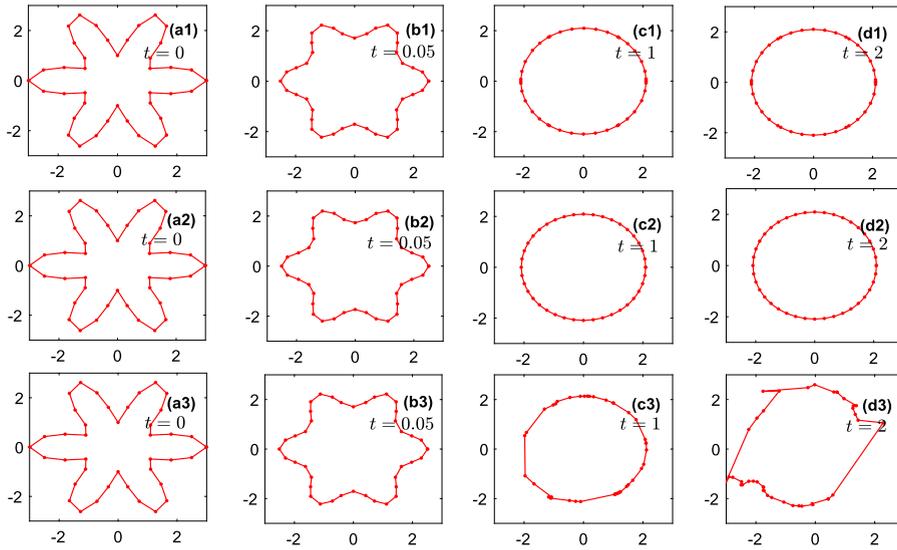


Fig. 7. Evolution of the curve driven by SDF starting with Shape 4 as initial data by using the BGN2 scheme (shown in the first row), the equi-BGN2 scheme (shown in the second row) and without mesh regularization procedure (shown in the third row). The simulations are conducted with a grid number of $N = 40$ and a time step size $\tau = 1/180$.

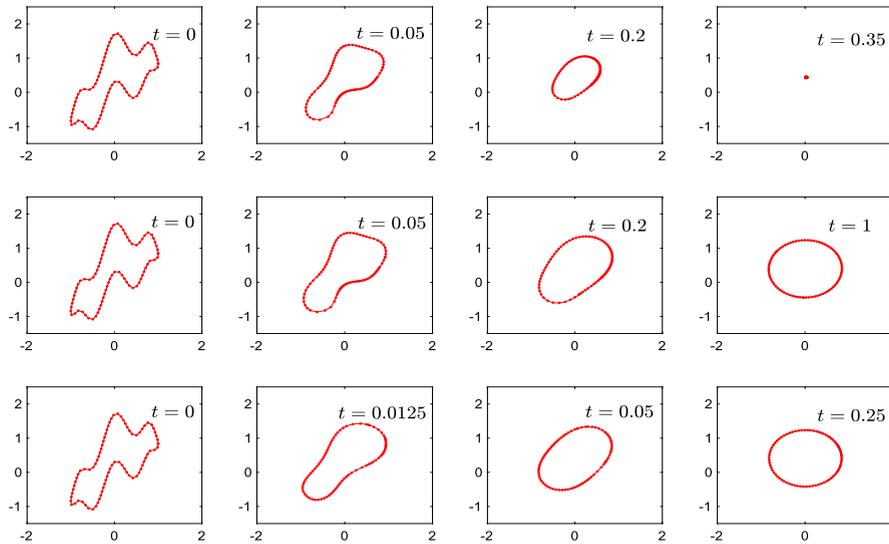


Fig. 8. Snapshots of the curve evolution using the proposed BGN2 schemes for three distinct geometric flows: CSF (first row), AP-CSF (second row) and SDF (third row). The simulations are conducted with $N = 80$ and $\tau = 1/640$.

Finally, we close this section by simulating the evolution of a nonconvex initial curve [3,27,29] driven by CSF, AP-CSF and SDF using the BGN2 schemes. The initial curve can be parametrized as

$$\mathbf{X}(\rho) = (\cos(2\pi\rho), \sin(\cos(2\pi\rho)) + \sin(2\pi\rho)(0.7 + \sin(2\pi\rho)\sin^2(6\pi\rho))),$$

for $\rho \in \mathbb{I} = [0, 1]$. The numerical results are depicted in Fig. 8. As shown in this figure, the CSF initially transforms the intricate curve into a circle before it disappears. Both the AP-CSF and SDF drive the curve to evolve into a perfect circle as its equilibrium shape.

6. Conclusions

We proposed two novel temporal second-order, BGN-based parametric finite element methods (i.e., the BGN2 and the equi-BGN2 schemes) for solving different geometric flows of curves (e.g., CSF, AP-CSF and SDF). Based on the BGN formulation and the corresponding semi-discrete FEM approximation, our numerical methods employ a Crank-Nicolson leap-frog method to discretize in time. The key idea lies in choosing a discrete inner product over the curve Γ^m , such that the time level t_m coincides with the time

at which all quantities have approximations with an error of $\mathcal{O}(\tau^2)$. We established the well-posedness of the BGN2 scheme under some suitable assumptions. Additionally, we showed that the equi-BGN2 scheme is unconditional energy-stable. We emphasized the use of shape metrics (manifold distance and Hausdorff distance) rather than function norms (e.g., L^2 -norm, L^∞ -norm) to measure numerical errors of BGN-based schemes. In the case of certain initial curves, such as a ‘flower’ shape, we found that the BGN2 scheme (resp. the equi-BGN2 scheme), in conjunction with the BGN1 scheme (resp. the equi-BGN1 scheme) for mesh regularization, exhibited remarkable stability in practical simulations. Extensive numerical experiments demonstrated that the proposed BGN2 and equi-BGN2 schemes achieve second-order accuracy in time, as measured by the shape metrics, outperforming the BGN1 scheme in terms of accuracy.

Furthermore, it is worth mentioning that the approach we have presented for constructing a temporal high-order BGN-based scheme can be readily extended to address various other problems, such as anisotropic geometric flows [5], Willmore flow [11], two-phase flow [20], solid-state dewetting [31] and geometric flows in 3D [30].

In our future research, we will further investigate the development of structure-preserving temporal high-order BGN-based schemes [7,22] and conduct the numerical analysis of the BGN-based schemes with respect to the shape metric. These investigations will contribute to enhancing the overall understanding and applicability of the BGN type scheme in different contexts.

CRedit authorship contribution statement

Wei Jiang: Conceptualization, Methodology, Supervision, Writing – review & editing. **Chunmei Su:** Conceptualization, Methodology, Supervision, Writing – review & editing. **Ganghui Zhang:** Investigation, Methodology, Software, Validation, Visualization, Writing – original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgement

We would like to thank the anonymous reviewers for their helpful comments and suggestions. This work was partially supported by the NSFC 12271414 and 11871384 (W. J.), the Natural Science Foundation of Hubei Province Grant No. 2022CFB245 (W. J.), and NSFC 12201342 (C. S. and G. Z.). The numerical calculations in this paper have been done on the supercomputing system in the Supercomputing Center of Wuhan University.

References

- [1] G. Aubert, P. Kornprobst, *Mathematical Problems in Image Processing*, Applied Mathematical Science, vol. 147, Springer, 2006.
- [2] Y. Bai, J. Yong, C. Liu, X. Liu, Y. Meng, Polyline approach for approximating Hausdorff distance between planar free-form curves, *Comput. Aided Des.* 43 (2011) 687–698.
- [3] M. Balažovjech, K. Mikula, A higher order scheme for a tangentially stabilized plane curve shortening flow with a driving force, *SIAM J. Sci. Comput.* 33 (2011) 2277–2294.
- [4] W. Bao, H. Garcke, R. Nürnberg, Q. Zhao, Volume-preserving parametric finite element methods for axisymmetric geometric evolution equations, *J. Comput. Phys.* 460 (2022) 111180.
- [5] W. Bao, W. Jiang, Y. Li, A symmetrized parametric finite element method for anisotropic surface diffusion of closed curves, *SIAM J. Numer. Anal.* 61 (2023) 617–641.
- [6] W. Bao, W. Jiang, Y. Wang, Q. Zhao, A parametric finite element method for solid-state dewetting problem with anisotropic surface energies, *J. Comput. Phys.* 330 (2017) 380–400.
- [7] W. Bao, Q. Zhao, A structure-preserving parametric finite element method for surface diffusion, *SIAM J. Numer. Anal.* 59 (2021) 2775–2799.
- [8] J.W. Barrett, H. Garcke, R. Nürnberg, A parametric finite element method for fourth order geometric evolution equations, *J. Comput. Phys.* 222 (2007) 441–467.
- [9] J.W. Barrett, H. Garcke, R. Nürnberg, On the variational approximation of combined second and fourth order geometric evolution equations, *SIAM J. Sci. Comput.* 29 (2007) 1006–1041.
- [10] J.W. Barrett, H. Garcke, R. Nürnberg, On the parametric finite element approximation of evolving hypersurfaces in \mathbb{R}^3 , *J. Comput. Phys.* 227 (2008) 4281–4307.
- [11] J.W. Barrett, H. Garcke, R. Nürnberg, Parametric approximation of Willmore flow and related geometric evolution equations, *SIAM J. Sci. Comput.* 31 (2008) 225–253.
- [12] J.W. Barrett, H. Garcke, R. Nürnberg, The approximation of planar curve evolutions by stable fully implicit finite element schemes that equidistribute, *Numer. Methods Partial Differ. Equ.* 27 (2011) 1–30.
- [13] J.W. Barrett, H. Garcke, R. Nürnberg, Parametric finite element method approximations of curvature driven interface evolutions, in: *Handb. Numer. Anal.*, vol. 21, Elsevier, Amsterdam, 2020, pp. 275–423.
- [14] E. Bänsch, P. Morin, R.H. Nochetto, A finite element method for surface diffusion: the parametric case, *J. Comput. Phys.* 203 (2005) 321–343.
- [15] K. Deckelnick, G. Dziuk, On the approximation of the curve shortening flow, in: *Calculus of Variations, Applications and Computations*, Pont-à-Mousson, 1994, in: *Pitman Res. Notes Math. Ser.*, vol. 326, Longman Sci. Tech., Harlow, UK, 1995, pp. 100–108.
- [16] K. Deckelnick, G. Dziuk, C.M. Elliott, Computation of geometric partial differential equations and mean curvature flow, *Acta Numer.* 14 (2005) 139–232.
- [17] G. Dziuk, Convergence of a semi-discrete scheme for the curve shortening flow, *Math. Models Methods Appl. Sci.* 04 (1994) 589–606.

- [18] G. Dziuk, Discrete anisotropic curve shortening flow, *SIAM J. Numer. Anal.* 36 (1999) 1808–1830.
- [19] C.M. Elliott, H. Fritz, On approximations of the curve shortening flow and of the mean curvature flow based on the DeTurck trick, *IMA J. Numer. Anal.* 37 (2016) 543–603.
- [20] H. Garcke, R. Nürnberg, Q. Zhao, Structure-preserving discretizations of two-phase Navier–Stokes flow using fitted and unfitted approaches, *J. Comput. Phys.* 489 (2023) 112276.
- [21] N. Hurl, W. Layton, Y. Li, M. Moraiti, The unstable mode in the Crank-Nicolson Leap-Frog method is stable, *Int. J. Numer. Anal. Model.* 13 (2016) 753–762.
- [22] W. Jiang, B. Li, A perimeter-decreasing and area-conserving algorithm for surface diffusion flow of curves, *J. Comput. Phys.* 443 (2021) 110531.
- [23] W. Jiang, C. Su, G. Zhang, A convexity-preserving and perimeter-decreasing parametric finite element method for the area-preserving curve shortening flow, *SIAM J. Numer. Anal.* 61 (2023) 1989–2010.
- [24] M. Kimura, Numerical analysis of moving boundary problems using the boundary tracking method, *Jpn. J. Ind. Appl. Math.* 14 (1997) 373–398.
- [25] B. Kovács, B. Li, C. Lubich, A convergent evolving finite element algorithm for mean curvature flow of closed surfaces, *Numer. Math.* 143 (2019) 797–853.
- [26] B. Kovács, B. Li, C. Lubich, A convergent algorithm for forced mean curvature flow driven by diffusion on the surface, *Interfaces Free Bound.* 22 (2020) 443–464.
- [27] J.A. Mackenzie, M. Nolan, C.F. Rowlatt, R.H. Insall, An adaptive moving mesh method for forced curve shortening flow, *SIAM J. Sci. Comput.* 41 (2019) A1170–A1200.
- [28] K. Mikula, D. Sevcovic, Evolution of plane curves driven by a nonlinear function of curvature and anisotropy, *SIAM J. Appl. Math.* 61 (2001) 1473–1501.
- [29] K. Mikula, D. Sevcovic, A direct method for solving an anisotropic mean curvature flow of plane curves with an external force, *Math. Methods Appl. Sci.* 27 (2004) 1545–1565.
- [30] Q. Zhao, W. Jiang, W. Bao, A parametric finite element method for solid-state dewetting problems in three dimensions, *SIAM J. Sci. Comput.* 42 (2020) B327–B352.
- [31] Q. Zhao, W. Jiang, W. Bao, An energy-stable parametric finite element method for simulating solid-state dewetting, *IMA J. Numer. Anal.* 41 (2021) 2026–2055.