SIAM J. SCI. COMPUT.
Vol. 46, No. 5, pp. A2874–A2898

STABLE BACKWARD DIFFERENTIATION FORMULA TIME DISCRETIZATION OF BGN-BASED PARAMETRIC FINITE ELEMENT METHODS FOR GEOMETRIC FLOWS*

WEI JIANG†, CHUNMEI SU‡, AND GANGHUI ZHANG§

Abstract. We propose a novel class of temporal high-order parametric finite element methods for solving a wide range of geometric flows of curves and surfaces. By incorporating the backward differentiation formula (BDF) for time discretization into the BGN formulation, originally proposed by Barrett, Garcke, and Nürnberg (*J. Comput. Phys.*, 222 (2007), pp. 441–467), we successfully develop high-order BGN/BDFk schemes. The proposed BGN/BDFk schemes not only retain almost all the advantages of the classical first-order BGN scheme such as computational efficiency and good mesh quality, but also exhibit the desired kth-order temporal accuracy in terms of shape metrics, ranging from second-order to fourth-order accuracy. Furthermore, we validate the performance of our proposed BGN/BDFk schemes through extensive numerical examples, demonstrating their high-order temporal accuracy for various types of geometric flows while maintaining good mesh quality throughout the evolution.

Key words. parametric finite element method, geometric flow, BGN scheme, backward differentiation formulae, high-order accuracy in time, good mesh quality

MSC codes. 74H15, 74S05, 74M15, 65M60

DOI. 10.1137/23M1625597

1. Introduction.

Geometric flows, which are also known as geometric PDEs, have been a subject of significant interest in the past several decades. These flows involve the evolution of a geometric shape from one form to another and have found applications in various fields, such as grain boundary motion [36], solid-state dewetting [5, 43, 44], image processing [14, 40], biomembranes [17], and cellular automata [25, 39]. Numerical simulation has played a crucial role in this rapidly growing research area, aiding in the understanding of the underlying theory and guiding experimental investigations.

One of the most important numerical methods is the parametric finite element method, which was first proposed by Dziuk [19] for simulating mean curvature flow of surfaces in three-dimensional space. Since then, this type of numerical scheme has been extensively employed for solving various types of geometric flows arising from science and engineering problems, including mean curvature flow [20, 30, 31], surface diffusion flow [6, 27], Willmore flow [22, 23, 32], anisotropic flow [4, 9, 21], and generalized mean curvature flow [8, 37, 13]. For a more comprehensive overview

*Submitted to the journal's Numerical Algorithms for Scientific Computing section December 19, 2023; accepted for publication (in revised form) June 26, 2024; published electronically September 6, 2024.

https://doi.org/10.1137/23M1625597

Funding: The work of the first author is supported by NSFC 12271414 and 11871384, the work of the second and third authors were supported by National Key R&D Program of China (2023YFA1008902), and NSFC 12201342. This work is also supported by the Center of High Performance Computing, Tsinghua University.

†School of Mathematics and Statistics, Hubei Key Laboratory of Computational Science, Wuhan University, Wuhan 430072, China (jiangwei1007@whu.edu.cn).

‡Corresponding author. Yau Mathematical Sciences Center, Tsinghua University, Beijing 100084, China (sucm@tsinghua.edu.cn).

§Yau Mathematical Sciences Center, Tsinghua University, Beijing 100084, China (gh-zhang19@mails.tsinghua.edu.cn).

of recent advances in the parametric finite element method, we recommend referring to the survey papers [12, 16].

One of the main difficulties in solving the geometric flows by parametric finite element methods is the mesh distortion problem. As time evolves, the nodes may cluster together and the mesh deteriorates, leading to unstable simulations and algorithm failures. To date, numerous efforts have been made to address this issue in the literature. One approach is the use of artificial mesh regularization methods to improve the distribution of mesh points during the evolution, as proposed by Bänsch, Morin, and Nochetto [3]. Another approach is to include an additional tangential velocity functional in the equation to prevent numerical solutions from forming various instabilities, as studied by Mikula and Ševčovič [41, 35]. As demonstrated in [15, 24], DeTurck's trick or, more precisely, harmonic map meat flows can be utilized to maintain good mesh quality. In addition, a reparametrization technique based on a discrete harmonic map [42] has been employed for remeshing the evolutionary polyhedron at each time step. Recently, Hu and Li [26] have proposed a new evolving surface finite element method by introducing an artificial tangential velocity to improve the mesh quality for mean curvature flow and Willmore flow.

Instead of utilizing the mesh redistribution approach, the so-called BGN scheme, originally proposed by Barrett, Garcke, and Nürnberg, is a parametric finite element method constructed based on a formulation which allows an intrinsic tangential velocity to ensure a good distribution of mesh points [7, 8, 11]. Taking curve shortening flow (CSF) as an example, now we present the key idea of the BGN scheme and our approach to developing high-order BGN-based schemes. Let $\Gamma := \Gamma(t)$ be a family of simple closed curves embedded in the two-dimensional plane driven by CSF, i.e., the velocity is given by

$$\mathcal{V} = -\kappa \mathbf{n}. \tag{1.1}$$

Here κ represents the curvature of the curve and we always assume that a circle has a positive curvature, and $\mathbf{n}$ is the outward unit normal to Γ. First, the curve $\Gamma(t)$ can be represented by a vector function $\mathbf{X}(\cdot, t) : \mathbb{I} \to \mathbb{R}^2$, where $\mathbb{I} := \mathbb{R}/\mathbb{Z}$ is the periodic interval $[0, 1]$. Then, (1.1) is rewritten as the following BGN formulation [8],

$$\begin{aligned} \partial_t \mathbf{X} \cdot \mathbf{n} &= -\kappa, \\ \kappa \mathbf{n} &= -\partial_{ss} \mathbf{X}, \end{aligned} \tag{1.2}$$

where $s := s(t)$ represents the arc length of $\Gamma(t)$. Compared to the original flow (1.1), a new variable κ is introduced. This formulation is attractive since the normal velocity remains unchanged, preserving the shape of the evolving curve. On the other hand, the tangential velocity is not prescribed in (1.2), allowing for an intrinsic tangential movement. Based on this formulation, a semidiscrete scheme was proposed [8],

$$\begin{aligned} \frac{\mathbf{X}^{m+1} - \mathbf{X}^m}{\tau} \cdot \mathbf{n}^m &= -\kappa^{m+1}, \\ \kappa^{m+1} \mathbf{n}^m &= -\partial_{s^m s^m} \mathbf{X}^{m+1}, \end{aligned} \tag{1.3}$$

where $\cdot^m$ represents the approximation solution at time $t_m := m\tau$ with τ as the time step size, e.g., Γ^m (which is parametrized by $\mathbf{X}^m$) is an approximation of $\Gamma(t_m)$, and $\mathbf{n}^m$, s^m correspond to the approximations of the unit outer normal vector and arc length, respectively. By using the linear finite element method in space and taking the variational formulation over the polygon Γ^m on both sides of equations, Barrett, Garcke, and Nürnberg derived the classical BGN scheme for CSF. It was shown that

the resulting fully discrete BGN scheme is well-posed and possesses several desirable properties, such as unconditional stability, energy dissipation, and asymptotic long-time mesh equal distribution [7, 8, 12]. However, it is worth noting that the fully discrete BGN scheme is limited to first-order accuracy in time, and developing a high-order BGN-based scheme remains a challenging task. Actually, naive high-order time discretizations based on the BGN formulation might cause mesh distortion problems [18, 28]. For example, as pointed out in [18], high-order time discretizations by the BDF, based on the BGN methods, probably become unstable (see Figures 4–5).

In this paper, we present a series of temporal high-order schemes using the BDF based on the BGN formulation. The numerical instability problem pointed in [18] can be avoided effectively by carefully selecting the prediction curve/surface to integrate on. Taking the BDF2 time discretization of (1.2) as an example, we consider the following semi-discrete in time scheme for CSF:

$$\begin{aligned} \frac{\frac{3}{2}\mathbf{X}^{m+1}-2\mathbf{X}^{m}+\frac{1}{2}\mathbf{X}^{m-1}}{\tau}\widetilde{\mathbf{n}}^{m+1} &= -\kappa^{m+1}, \\ \kappa^{m+1}\widetilde{\mathbf{n}}^{m+1} &= -\partial_{\tilde{s}^{m+1}\tilde{s}^{m+1}}\mathbf{X}^{m+1}. \end{aligned} \tag{1.4}$$

Here, we utilize a semi-implicit approach to avoid a fully implicit scheme. Thus a suitable explicit approximation $\widetilde{\Gamma}^{m+1}$ of $\Gamma(t_{m+1})$, on which $\widetilde{\mathbf{n}}^{m+1}$, $\tilde{s}^{m+1}$ are then explicitly calculated, is required such that only linear algebraic equations need to be solved at each time step. This approach maintains high-order accuracy by carefully selecting the approximation of the integration curve $\widetilde{\Gamma}^{m+1}$ to adjust all numerical quantities at the same time level t_{m+1}. *Indeed, we emphasize that the selection of* $\widetilde{\Gamma}^{m+1}$ *is key to the success of this scheme, especially in maintaining good mesh distribution and stability during the evolution. Instead of using the classical extrapolation formulae from the former parametrized functions* $\mathbf{X}^m$ *and* $\mathbf{X}^{m-1}$, *which might lead to mesh distortion problems* [18], *here we choose the prediction curve* $\widetilde{\Gamma}^{m+1}$ *(or equivalently* $\widetilde{\mathbf{X}}^{m+1}$*) as the solution of the classical first-order BGN scheme* (1.3). *Extensive numerical experiments indicate that expected second-order accuracy in time can be achieved in terms of shape metrics, good mesh quality is maintained during the evolution and mesh distortion can be prevented effectively (see section* 3 *and Remark* 4.3). The same idea can be further extended to develop BGN/BDFk schemes for $k = 3, 4, 5, 6$.

It is worth mentioning that various efforts have been made in the literature to develop high-order temporal schemes for solving geometric flows, each based on different approaches. For example, an implicit Crank-Nicolson-type scheme was designed for forced CSF by combining with mesh redistribution [2] or an adaptive moving mesh technique [34] to maintain a good mesh quality. Both schemes have been shown to converge quadratically in time but require solving a system of nonlinear equations at each time step. Based on evolving surface finite element methods and the BDF, Kovács, Li, and Lubich proposed some high-order numerical schemes for solving mean curvature flow and Willmore flow [30, 31, 32]. Due to the lack of the tangential velocity, these schemes may suffer from mesh clustering and distortion, leading to breakdowns of simulations in some cases. Very recently, by introducing an artificial tangential velocity determined by a harmonic map from a fixed reference surface to the unknown evolving surface, Duan and Li have proposed a new class of parametric finite element methods, including a second-order scheme, with good mesh quality for simulating various types of geometric flows [18].

Among the aforementioned works, our proposed BGN-based high-order schemes inherit most of the advantages from the classical BGN scheme, including

- good mesh quality is maintained during the evolution and no numerical instability occurs;

- the methods can be easily implemented very efficiently, as only linear algebraic equations need to be solved at each time step;
- the methods can be extended straightforwardly to a wide range of geometric flows of curves or surfaces, such as area-preserving mean curvature flow, generalized mean curvature flow, and Willmore flow;

and, more importantly,

- the approach can be extended to higher-order BDFk schemes which converge at the k-th order in time in terms of shape metrics while keeping all the above superiorities.

Compared to the method proposed by [18] that also employs BDF methods and achieves favorable mesh quality, we emphasize that our approach is based on the classical BGN scheme, requiring only minor modifications. Moreover, our innovative strategy of iteratively selecting the prediction curve/polyhedron using low-order BDFk methods effectively combines both the evolving nature of the problem and the desirable mesh properties. This strategy is promising for potential applications in other numerical methods for geometric flows.

The rest of this paper is organized as follows. In section 2, we provide a brief overview of the classical first-order BGN scheme, using CSF and mean curvature flow (MCF) as examples. In section 3, we propose high-order BGN/BDFk schemes for solving various types of geometric flows. To demonstrate the accuracy, efficiency, and applicability of our high-order algorithms, we present numerous numerical examples for simulating curve and surface evolution driven by different types of geometric flows in section 4. Finally, we summarize our findings, draw some conclusions based on the results, and discuss potential future research directions in this field in section 5.

2. Review of classical BGN scheme. In this section, we review the classical first-order BGN schemes for CSF and its three-dimensional analogue MCF, which were proposed by Barrett, Garcke and Nürnberg [7, 8, 10, 12]. To begin with, we rewrite the CSF into the BGN formulation (1.2).

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 ρ_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 the grid. Define the linear finite element space as

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

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

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

where $\mathbf{X}^h$ is a parametrization of Γ^h, $\mathbf{X}^h(\rho_j)$ is the vertex of the polygon Γ^h, and u,v are two scalar/vector piecewise continuous functions with possible jumps at the nodes $\{\rho_j\}_{j=1}^N$ and $u(\rho_j^{\pm})=\lim_{\rho\to\rho_j^{\pm}}u(\rho)$.

Subsequently, the semidiscrete scheme of the formulation (1.2) is as follows: given an initial polygon $\Gamma^h(0)$ with vertices lying on the initial curve $\Gamma(0)$ in a clockwise manner, parametrized by $\mathbf{X}^h(\cdot,0)\in[V^h]^2$, find $(\mathbf{X}^h(\cdot,t),\kappa^h(\cdot,t))\in[V^h]^2\times V^h$ so that

$$\begin{cases}\left(\partial_t\mathbf{X}^h\cdot\mathbf{n}^h,\varphi^h\right)^h_{\Gamma^h}+\left(\kappa^h,\varphi^h\right)^h_{\Gamma^h}=0\quad\forall\ \varphi^h\in V^h,\\ \left(\kappa^h,\mathbf{n}^h\cdot\boldsymbol{\omega}^h\right)^h_{\Gamma^h}-\left(\partial_s\mathbf{X}^h,\partial_s\boldsymbol{\omega}^h\right)_{\Gamma^h}=0\quad\forall\ \boldsymbol{\omega}^h\in[V^h]^2,\end{cases}\tag{2.1}$$

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

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

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 (2.1) will always equidistribute the vertices along Γ^h for $t > 0$ if they are not locally parallel (see Remark 2.4 in [7]).

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 $m = 0, 1, 2, \ldots$, 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 $j = 1, \ldots, N \ \forall\ m > 0$. The discrete unit normal vector $\mathbf{n}^m$, the discrete inner product $(\cdot, \cdot)^h_{\Gamma^m}$, and the discrete operator ∂_s are defined similarly as in the semidiscrete case. Barrett, Garcke, and Nürnberg used a formal first-order approximation [7, 8] 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 semiimplicit BGN scheme reads as follows:

*BGN*1*, the classical first-order 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

$$\begin{cases} \left(\dfrac{\mathbf{X}^{m+1} - \mathbf{X}^m}{\tau}, \varphi^h \mathbf{n}^m\right)^h_{\Gamma^m} + \left(\kappa^{m+1}, \varphi^h\right)^h_{\Gamma^m} = 0 \quad \forall\ \varphi^h \in V^h, \\ \left(\kappa^{m+1}, \mathbf{n}^m \cdot \boldsymbol{\omega}^h\right)^h_{\Gamma^m} - \left(\partial_s \mathbf{X}^{m+1}, \partial_s \boldsymbol{\omega}^h\right)_{\Gamma^m} = 0 \quad \forall\ \boldsymbol{\omega}^h \in [V^h]^2. \end{cases} \tag{2.2}$$

The well-posedness and energy stability were established under some mild conditions [12]. In practice, numerous numerical results show that the classical BGN scheme (2.2) converges quadratically in space [8] and linearly in time [28].

The above idea has been successfully extended to the MCF in $\mathbb{R}^3$ [10]. The governing equation of MCF is given by

$$\mathcal{V} = -\mathcal{H}\,\mathbf{n}, \tag{2.3}$$

where $\mathcal{H}$ is the mean curvature of the hypersurface. Following the lines in [10, 12], (2.3) can be reformulated as

$$\begin{aligned} \partial_t \mathbf{X} \cdot \mathbf{n} &= -\mathcal{H}, \\ \mathcal{H}\mathbf{n} &= -\Delta_\Gamma \mathrm{Id}, \end{aligned} \tag{2.4}$$

where Δ_Γ is the surface Laplacian (i.e., Laplace–Beltrami operator) and Id is the identity map on Γ. For the finite element approximation, suppose we have a polyhedra surface Γ^m approximating the closed surface $\Gamma(t_m)$, which is a union of nondegenerate triangles with no hanging vertices

$$\Gamma^m := \bigcup_{j=1}^{J} \overline{\sigma_j^m},$$

where $\{\sigma_j^m\}_{j=1}^J$ is a family of mutually disjoint open triangles. Set $h = \max_{1\le j\le J} \operatorname{diam}(\sigma_j^m)$. Denote $W_m^h := W^h(\Gamma^m)$ by the space of scalar continuous piecewise linear functions on Γ^m, i.e.,

$$W_m^h = W^h(\Gamma^m) := \left\{ u \in C(\Gamma^m, \mathbb{R}) :\ u|_{\sigma_j^m} \text{ is linear } \forall j = 1, \dots, J \right\} \subseteq H^1(\Gamma^m, \mathbb{R}).$$

Similarly, for piecewise continuous scalar or vector functions $u, v \in L^2(\Gamma^m, \mathbb{R}^3)$ with possible jumps at the edges of σ_j^m, the L^2 inner product $(\cdot,\cdot)_{\Gamma^m}$ over the polyhedral surface Γ^m,

$$(u, v)_{\Gamma^m} = \int_{\Gamma^m} u \cdot v \, \mathrm{d}A,$$

can be approximated by the mass lumped inner product

$$(u, v)_{\Gamma^m}^h := \frac{1}{3} \sum_{j=1}^{J} |\sigma_j^m| \sum_{k=1}^{3} (u \cdot v) \left((\mathbf{q}_{j_k}^m)^- \right),$$

where $\{\mathbf{q}_{j_1}, \mathbf{q}_{j_2}, \mathbf{q}_{j_3}\}$ are the vertices of the triangle σ_j^m, $|\sigma_j^m|$ is the area of σ_j^m, and

$$u\left((\mathbf{q}_{j_k}^m)^-\right) = \lim_{\sigma_j^m \ni \mathbf{x} \to \mathbf{q}_{j_k}^m} u(\mathbf{x}).$$

Similarly, the classical first-order BGN scheme for MCF was proposed as the following [10]:

BGN1, the classical BGN, first-order scheme for MCF: Given Γ^0 and the identity function $\mathbf{X}^0 \in W_0^h$ on Γ^0, for $m \ge 0$, find $\mathbf{X}^{m+1} \in [W_m^h]^3$ and $\mathcal{H}^{m+1} \in W_m^h$ such that

$$\text{(2.5)} \qquad \begin{cases} \left(\dfrac{\mathbf{X}^{m+1} - \mathbf{X}^m}{\tau}, \varphi^h \mathbf{n}^m \right)_{\Gamma^m}^h + \left(\mathcal{H}^{m+1}, \varphi^h \right)_{\Gamma^m}^h = 0 \quad \forall\ \varphi^h \in W_m^h, \\ \left(\mathcal{H}^{m+1}, \mathbf{n}^m \cdot \boldsymbol{\omega}^h \right)_{\Gamma^m}^h - \left(\nabla_{\Gamma^m} \mathbf{X}^{m+1}, \nabla_{\Gamma^m} \boldsymbol{\omega}^h \right)_{\Gamma^m} = 0 \quad \forall\ \boldsymbol{\omega}^h \in [W_m^h]^3, \end{cases}$$

where the outward unit normal vector $\mathbf{n}^m$ of Γ^m is defined piecewisely over triangle $\{\sigma_j^m\}$ as

$$\mathbf{n}^m|_{\sigma_j^m} = \mathbf{n}_j^m := \frac{(\mathbf{q}_{j_2}^m - \mathbf{q}_{j_1}^m) \times (\mathbf{q}_{j_3}^m - \mathbf{q}_{j_1}^m)}{|(\mathbf{q}_{j_2}^m - \mathbf{q}_{j_1}^m) \times (\mathbf{q}_{j_3}^m - \mathbf{q}_{j_1}^m)|},$$

and $\mathbf{X}^m(\cdot)$ is the identity function on $[W_m^h]^3$. The surface gradient ∇_Γ over polyhedra Γ is defined for $f \in W^h(\Gamma)$ piecewisely on triangles $\{\sigma_j\}$ with vertices $\{\mathbf{q}_{j_1}, \mathbf{q}_{j_2}, \mathbf{q}_{j_3}\}$ as

$$(\nabla_\Gamma f)|_{\sigma_j} := f(\mathbf{q}_{j_1}) \frac{(\mathbf{q}_{j_3} - \mathbf{q}_{j_2}) \times \mathbf{n}_j}{|\sigma_j|} + f(\mathbf{q}_{j_2}) \frac{(\mathbf{q}_{j_1} - \mathbf{q}_{j_3}) \times \mathbf{n}_j}{|\sigma_j|} + f(\mathbf{q}_{j_3}) \frac{(\mathbf{q}_{j_2} - \mathbf{q}_{j_1}) \times \mathbf{n}_j}{|\sigma_j|}.$$

The well-posedness and energy stability were also established under some mild conditions [10], and a quadratic convergence rate in space was reported.

Throughout the paper, the first-order BGN scheme (2.2) or (2.5) is referred to as the BGN1 scheme.

3. High order in time, BGN-based algorithms. In this section, we propose high-order temporal schemes based on the BDF. For simplicity of notation, here we only present the schemes for the CSF, and similar schemes can be proposed for MCF in $\mathbb{R}^3$. Specifically, we approximate the term $\partial_t \mathbf{X}$ based on the following Taylor expansions [33]:

$$\partial_t \mathbf{X}(\cdot,t_{m+1}) = \begin{cases} \frac{\frac{3}{2}\mathbf{X}(\cdot,t_{m+1})-2\mathbf{X}(\cdot,t_m)+\frac{1}{2}\mathbf{X}(\cdot,t_{m-1})}{\tau} + \mathcal{O}(\tau^2), \\ \frac{\frac{11}{6}\mathbf{X}(\cdot,t_{m+1})-3\mathbf{X}(\cdot,t_m)+\frac{3}{2}\mathbf{X}(\cdot,t_{m-1})-\frac{1}{3}\mathbf{X}(\cdot,t_{m-2})}{\tau} + \mathcal{O}(\tau^3), \\ \frac{\frac{25}{12}\mathbf{X}(\cdot,t_{m+1})-4\mathbf{X}(\cdot,t_m)+3\mathbf{X}(\cdot,t_{m-1})-\frac{4}{3}\mathbf{X}(\cdot,t_{m-2})+\frac{1}{4}\mathbf{X}(\cdot,t_{m-3})}{\tau} + \mathcal{O}(\tau^4). \end{cases}$$

Thus the velocity is approximated with an error of $\mathcal{O}(\tau^k)$, $2 \le k \le 4$, at the time level t_{m+1}. By taking the mass lumped inner product over a suitable predictor $\widetilde{\Gamma}^{m+1}$, which is an approximation of $\Gamma(t_{m+1})$, we obtain the following high-order schemes (denoted as BGN/BDFk schemes) for $2 \le k \le 4$.

BGN/BDFk, high-order schemes for CSF: For $k = 2,3,4$, $m \ge k-1$, find $\mathbf{X}^{m+1} \in [V^h]^2$ and $\kappa^{m+1} \in V^h$ such that

$$\begin{cases} \left(\dfrac{a\mathbf{X}^{m+1} - \widehat{\mathbf{X}}^m}{\tau}, \varphi^h \widetilde{\mathbf{n}}^{m+1} \right)^h_{\widetilde{\Gamma}^{m+1}} + \left(\kappa^{m+1}, \varphi^h\right)^h_{\widetilde{\Gamma}^{m+1}} = 0 \quad \forall\ \varphi^h \in V^h, \\ \left(\kappa^{m+1}, \widetilde{\mathbf{n}}^{m+1} \cdot \boldsymbol{\omega}^h\right)^h_{\widetilde{\Gamma}^{m+1}} - \left(\partial_s \mathbf{X}^{m+1}, \partial_s \boldsymbol{\omega}^h\right)_{\widetilde{\Gamma}^{m+1}} = 0 \quad \forall\ \boldsymbol{\omega}^h \in [V^h]^2, \end{cases} \tag{3.1}$$

where a, $\widehat{\mathbf{X}}^m$ are defined as

$$\text{BDF2}: a = \frac{3}{2}, \qquad \widehat{\mathbf{X}}^m = 2\mathbf{X}^m - \frac{1}{2}\mathbf{X}^{m-1}; \tag{3.2}$$

$$\text{BDF3}: a = \frac{11}{6}, \qquad \widehat{\mathbf{X}}^m = 3\mathbf{X}^m - \frac{3}{2}\mathbf{X}^{m-1} + \frac{1}{3}\mathbf{X}^{m-2}; \tag{3.3}$$

$$\text{BDF4}: a = \frac{25}{12}, \qquad \widehat{\mathbf{X}}^m = 4\mathbf{X}^m - 3\mathbf{X}^{m-1} + \frac{4}{3}\mathbf{X}^{m-2} - \frac{1}{4}\mathbf{X}^{m-3}, \tag{3.4}$$

where $\widetilde{\Gamma}^{m+1}$, described by $\widetilde{\mathbf{X}}^{m+1} \in [V^h]^2$, is a suitable approximation of $\Gamma(t_{m+1})$, $\widetilde{\mathbf{n}}^{m+1} := -(\frac{\partial_\rho \widetilde{\mathbf{X}}^{m+1}}{|\partial_\rho \widetilde{\mathbf{X}}^{m+1}|})^\perp$ is the normal vector, and the derivative ∂_s is defined with respect to the arc length of $\widetilde{\Gamma}^{m+1}$.

Before introducing the specific choice of the approximation $\widetilde{\Gamma}^{m+1}$, we first establish the fundamental properties of the above BGN/BDFk schemes (3.1). The first crucial property is the well-posedness of (3.1) under some mild conditions.

THEOREM 3.1 (well-posedness). *For $k = 2,3,4$, $m \ge k-1$, assume that the polygon $\widetilde{\Gamma}^{m+1}$ in the BGN/BDFk schemes* (3.1) *satisfies the following two conditions:*
(i) *At least two vectors in $\{\widetilde{\mathbf{h}}_j^{m+1}\}_{j=1}^N$ are not parallel, i.e.,*

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

(ii) *No vertices degenerate on $\widetilde{\Gamma}^{m+1}$, i.e.,*

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

Then the above BGN/BDFk schemes (3.1) *are well-posed, i.e., there exists a unique solution $(\mathbf{X}^{m+1}, \kappa^{m+1}) \in [V^h]^2 \times V^h$ of* (3.1).

Proof. Thanks to the linearity of the scheme (3.1), it suffices to prove that the following algebraic system for $(\mathbf{X},\kappa)\in[V^h]^2\times V^h$ has only a zero solution,

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

Taking $\varphi^h=\kappa$ and $\boldsymbol{\omega}^h=\mathbf{X}$, noticing that $a>0$, we arrive at

$$\mathbf{X}\equiv\mathbf{X}^c,\qquad \kappa\equiv\kappa^c. \tag{3.5}$$

Then the standard argument in [7, Theorem 2.1] yields $\mathbf{X}^c=0$ and $\kappa^c=0$ by the assumption on $\widetilde{\Gamma}^{m+1}$. ∎

It remains to determine $\widetilde{\Gamma}^{m+1}$ or, equivalently, $\widetilde{\mathbf{X}}^{m+1}$. Indeed, the formulation of this predictor plays a crucial role in the success of BGN/BDFk schemes. A natural approach is to apply standard extrapolation formulas, such as

$$\widetilde{\mathbf{X}}^{m+1}=2\mathbf{X}^m-\mathbf{X}^{m-1}, \qquad \text{BDF2}, \tag{3.6}$$

$$\widetilde{\mathbf{X}}^{m+1}=3\mathbf{X}^m-3\mathbf{X}^{m-1}+\mathbf{X}^{m-2}, \qquad \text{BDF3}, \tag{3.7}$$

$$\widetilde{\mathbf{X}}^{m+1}=4\mathbf{X}^m-6\mathbf{X}^{m-1}+4\mathbf{X}^{m-2}-\mathbf{X}^{m-3}, \qquad \text{BDF4}. \tag{3.8}$$

Unfortunately, as discussed in [18], naively extrapolating the curves as functions may result in the instability of high-order schemes (see also Remark 4.3). In this paper, we utilize the solution of the lower-order BGN/BDFk scheme to predict the discrete polygon $\widetilde{\Gamma}^{m+1}$ or, equivalently, $\widetilde{\mathbf{X}}^{m+1}$, in the BGN/BDF$k$ scheme. In particular, for $k=1$, the BGN/BDF1 scheme represents the classical BGN1 scheme, which is used to predict $\widetilde{\Gamma}^{m+1}$ in the BGN/BDF2 scheme.

To start the BGN/BDFk scheme, it is necessary to prepare the initial data $\mathbf{X}^0,\dots,\mathbf{X}^{k-1}$, which are supposed to be approximations of $\mathbf{X}(\cdot,0),\dots,\mathbf{X}(\cdot,t_{k-1})$ with an error at the kth order $\mathcal{O}(\tau^k)$. This can be accomplished by utilizing the BGN1 scheme (2.2) with a finer time step. Specifically, to obtain an approximation of $\mathbf{X}(\cdot,t_1)$ with an error at $\mathcal{O}(\tau^k)$, where $k=2,3,4$, it is sufficient to implement the BGN1 scheme with a time step size $\widetilde{\tau}\sim\tau^{k-1}$ by $\tau/\widetilde{\tau}$ steps. Taking into account the truncation error of the BGN1 scheme (2.2), the accumulated temporal error at $t_1=\tau$ is given by

$$\widetilde{\tau}^2*\tau/\widetilde{\tau}\sim\tau^k,\quad k=2,3,4.$$

Now we are ready to present our BGN/BDFk schemes. Throughout all the algorithms, we always use Γ^m and $\widetilde{\Gamma}^m$ to represent the polygon described by the vector functions $\mathbf{X}^m,\widetilde{\mathbf{X}}^m\in[V^h]^2$, respectively.

Remark 3.2. A similar approach can be utilized to construct BGN/BDF5 and BGN/BDF6 algorithms, but for the sake of brevity, we omit the details here.

It is worth noting that the classical first-order BGN scheme exhibits favorable properties in terms of mesh distribution. As observed in [8, 10], the node points are moved tangentially, resulting in eventual equidistribution in practice. Furthermore, it has been demonstrated in previous studies [12, 44] that the mesh will eventually become evenly distributed if the solution has an equilibrium state. In other words, the mesh becomes asymptotically equidistributed when the solution has a nondegenerate equilibrium, such as in the case of area-preserving CSF or surface diffusion. Additionally, recent discoveries have highlighted the significant role of the BGN1 scheme in

Algorithm 3.1 BGN/BDF2 algorithm.

Require: An initial curve $\Gamma(0)$ approximated by a polygon Γ^0 with N vertices, described by $\mathbf{X}^0 \in [V^h]^2$, time step τ, and terminate time T satisfying $T/\tau \in \mathbb{N}$.

Calculate $\mathbf{X}^1$ by using BGN1 scheme (2.2) with Γ^0 and τ. Set $m = 1$.
while $m < T/\tau$, **do**
- Compute $\widetilde{\mathbf{X}}^{m+1}$ by using BGN1 scheme (2.2) with Γ^m and τ.
- Compute $\mathbf{X}^{m+1}$ by using the BGN/BDF2 scheme (3.1)–(3.2) with $\mathbf{X}^{m-1}, \mathbf{X}^m, \widetilde{\Gamma}^{m+1}$, and τ.

$m = m + 1$;
end while

Algorithm 3.2 BGN/BDF3 algorithm.

Require: An initial curve $\Gamma(0)$ approximated by a polygon Γ^0 (parametrized as $\mathbf{X}^0 \in [V^h]^2$) with N vertices, terminate time T, time step τ, and a finer time step $\widetilde{\tau} \sim \tau^2$.

Calculate $\mathbf{X}^1$ by using BGN1 scheme (2.2) with Γ^0 and $\widetilde{\tau}$ for $\tau/\widetilde{\tau}$ steps.
Calculate $\mathbf{X}^2$ by using the BGN/BDF2 algorithm (Algorithm 3.1) with $\mathbf{X}^0$, $\mathbf{X}^1$, and τ. Set $m = 2$.
while $m < T/\tau$, **do**
- Compute $\widetilde{\mathbf{X}}^{m+1}$ by using the BGN/BDF2 algorithm (Algorithm 3.1) with $\mathbf{X}^{m-1}, \mathbf{X}^m$, and τ.
- Compute $\mathbf{X}^{m+1}$ by using the BGN/BDF3 scheme (3.1) and (3.3) with $\mathbf{X}^{m-2}, \mathbf{X}^{m-1}, \mathbf{X}^m, \widetilde{\Gamma}^{m+1}$, and τ.

$m = m + 1$;
end while

Algorithm 3.3 BGN/BDF4 algorithm.

Require: An initial curve $\Gamma(0)$ approximated by a polygon Γ^0 (parametrized as $\mathbf{X}^0 \in [V^h]^2$) with N vertices, terminate time T, time step τ, and a finer time step $\widetilde{\tau} \sim \tau^3$.

Calculate $\mathbf{X}^1$ by using BGN1 scheme (2.2) with Γ^0 and $\widetilde{\tau}$ for $\tau/\widetilde{\tau}$ steps.
Calculate $\mathbf{X}^2$ by using BGN1 scheme (2.2) with Γ^1 and $\widetilde{\tau}$ for another $\tau/\widetilde{\tau}$ steps.
Calculate $\mathbf{X}^3$ by using the BGN/BDF3 algorithm (Algorithm 3.2) with $\mathbf{X}^0$, $\mathbf{X}^1$, $\mathbf{X}^2$, and τ. Set $m = 3$.
while $m < T/\tau$, **do**
- Compute $\widetilde{\mathbf{X}}^{m+1}$ by the BGN/BDF3 algorithm (Algorithm 3.2) with $\mathbf{X}^{m-2}, \mathbf{X}^{m-1}, \mathbf{X}^m$, and τ.
- Compute $\mathbf{X}^{m+1}$ by the BGN/BDF4 scheme (3.1) and (3.4) with $\mathbf{X}^{m-3}, \mathbf{X}^{m-2}, \mathbf{X}^{m-1}, \mathbf{X}^m, \widetilde{\Gamma}^{m+1}$, and τ.

$m = m + 1$;
end while

improving mesh quality when implementing a second-order BGN-type Crank–Nicolson leapfrog scheme occasionally [28]. It is worth emphasizing that the BGN1 scheme is utilized at every step of the mentioned BDFk algorithms, offering great promise in ensuring favorable mesh distribution (cf. section 4) and preventing mesh distortion or numerically induced self-intersections.

4. Numerical results. In this section, we provide numerous numerical examples to demonstrate the high-order convergence and advantageous properties in terms of mesh distribution of our BGN/BDFk algorithms for various types of geometric flows.

4.1. Curve evolution. In this subsection, we mainly focus on the following types of geometric flows of curves in the plane:

- (CS), which is the L^2-gradient flow of the length functional $E(\Gamma) = \int_\Gamma \mathrm{d}s$.
- Area-preserving curve shortening flow (AP-CSF), which is the L^2-gradient flow of the length functional with the constraint of a fixed enclosed area. This is a nonlocal second-order geometric flow with the governing equation

$$\mathcal{V} = (-\kappa + \langle\kappa\rangle)\,\mathbf{n}, \tag{4.1}$$

  where $\langle\kappa\rangle := \int_{\Gamma(t)} \kappa\,\mathrm{d}s / \int_{\Gamma(t)} \mathrm{d}s$ is the average curvature.
- Generalized MCF (G-MCF), which is given by

$$\mathcal{V} = -\beta\kappa^\alpha\,\mathbf{n}, \tag{4.2}$$

  where α and β are two real numbers satisfying $\alpha\beta > 0$. For $0 < \alpha \neq 1$ and $\beta = 1$, it is called as the powers of MCF; for $\alpha = -1$ and $\beta = -1$, it is called as the inverse MCF.
- Willmore flow (WF), which is the L^2-gradient flow of the Willmore energy functional $W(\Gamma) = \frac{1}{2}\int_\Gamma \kappa^2\,\mathrm{d}s$. This is a fourth-order geometric flow with the governing equation

$$\mathcal{V} = \left(\partial_{ss}\kappa - \frac{1}{2}\kappa^3\right)\mathbf{n}. \tag{4.3}$$

Based on the wide applicability of BGN-type methods [7, 8, 9, 10, 12], we can easily extend our BGN/BDFk algorithms to the aforementioned geometric flows with minor adjustments.

To test the convergence rate of our proposed BGN/BDFk schemes, it is advisable to employ shape metrics for measuring the numerical errors of the BGN-type schemes, as they allow an intrinsic tangential movement in order to facilitate a more uniform distribution of mesh points along the evolving curve or surface (see [28, section 3]). In the following examples, we utilize the manifold distance to quantify the difference between two curves. Specifically, the manifold distance between two curves Γ_1 and Γ_2 is defined as [28, 44]

$$\mathrm{M}(\Gamma_1,\Gamma_2) := |\Omega_1 \Delta \Omega_2| = |\Omega_1| + |\Omega_2| - 2|\Omega_1 \cap \Omega_2|, \quad \Omega_1\Delta\Omega_2 = (\Omega_1 \setminus \Omega_2) \cup (\Omega_2 \setminus \Omega_1), \tag{4.4}$$

where Ω_1 and Ω_2 represent the regions enclosed by the two curves Γ_1 and Γ_2, respectively, and $|\Omega|$ denotes the area of Ω. It has been proved that the manifold distance satisfies the properties of symmetry, positivity, and triangle inequality (see Proposition 5.1 in [44]). Thus it can be seen as a type of shape metric. Under some suitable assumptions, it is related to the L^1 norm of a distance function. Specifically, assume that Γ_1 is a C^2 curve and $\Gamma_\delta := \{x \in \mathbb{R}^2 \,|\, |\mathrm{d}(x,\Gamma_1)| < \delta\}$ is its tabular neighborhood [16], where $\mathrm{d}(x,\Gamma_1)$ is the signed distance function. If $\Gamma_2 \subset \Gamma_\delta$ and the projection map from Γ_2 to Γ_1 is one-to-one, then the manifold distance is indeed the L^1 norm of the distance function

$$\mathrm{M}(\Gamma_1,\Gamma_2) = \int_{\Gamma_1} |\,\mathrm{d}(\cdot,\Gamma_2)\,|\,\mathrm{d}s.$$

Very recently, Bai and Li [1] have given a proof of convergence for Dziuk's parametric finite element method with finite elements of degree $k \geq 3$ for MCF. Their convergence result holds in the L^2 norm, which measures the L^2 norm of the distance between two curves or surfaces. We also note that since numerical solutions are here represented as polygonal curves/polyhedron surfaces, it is very easy to calculate the area/volume of the symmetric difference region, i.e., the manifold distance. In practice, the computation of manifold distance can be conveniently performed using the MATLAB packages `polyshape` and `alphaShape`.

To test the temporal errors, we fix the number of nodes N large enough so that the spatial error can be neglected, and we vary the time step. The numerical error and the corresponding convergence order at time T under the manifold distance are then computed as follows,

$$\mathcal{E}_\tau(T) = \mathcal{E}_M(T) = \mathrm{M}(\mathbf{X}^{T/\tau}, \mathbf{X}_{\mathrm{ref}}), \quad \mathrm{Order} = \log\left(\frac{\mathcal{E}_{\tau_1}(T)}{\mathcal{E}_{\tau_2}(T)}\right) \Big/ \log\left(\frac{\tau_1}{\tau_2}\right), \tag{4.5}$$

where $\mathbf{X}^{T/\tau}$ represents the computed solutions using a time step of τ until time T, and $\mathbf{X}_{\mathrm{ref}}$ represents the reference solutions, either given by the exact solution or approximated by some numerical solution with a sufficiently fine mesh size and tiny time step when the true solution is not available.

To test the stability properties of our proposed schemes, we investigate the evolution of the following three important quantities with respect to evolving curves: (1) the relative area loss $\Delta A(t)$, (2) the normalized perimeter $L(t)/L(0)$, and (3) the mesh distribution function $\Psi(t)$, which are defined, respectively, as follows:

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

where A^m and L^m represent the enclosed area and the perimeter of the polygon determined by $\mathbf{X}^m$, respectively.

Example 4.1 (convergence rate of the BGN/BDFk scheme for CSF). We check the convergence rate of the classical BGN1 scheme (2.2) and BGN/BDFk schemes (3.1) for CSF. The initial curve is chosen as either a unit circle or an ellipse defined by $x^2/4 + y^2 = 1$. The parameters are set as $N = 10000$, $T = 0.25$.

For the case of the unit circle, it is well known that the CSF admits the exact solution

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

While for the ellipse case, since the true solution is unavailable, we compute the reference solution using the BGN/BDF4 algorithm with a fine mesh including $N = 10000$ nodes and a tiny time step of $\tau = 1/2560$. Figures 1(a)–1(b) present a log-log plot of the numerical errors at time $T = 0.25$ for the classical BGN1 scheme (2.2) and BGN/BDFk schemes (3.1). It can be clearly observed that the classical BGN1 scheme has only first-order accuracy in time, while the BGN/BDFk scheme achieves kth-order accuracy with $2 \leq k \leq 4$ for both the unit circle and ellipse cases. Furthermore, Figures 1(c) and 1(d) show the evolution of the mesh distribution function $\Psi(t)$ of BGN/BDFk algorithms, from which we clearly see that the proposed BGN/BDFk schemes share the same favorable properties with respect to the mesh distribution. Once the initial curve is approximated by a polygon with high mesh quality, the mesh

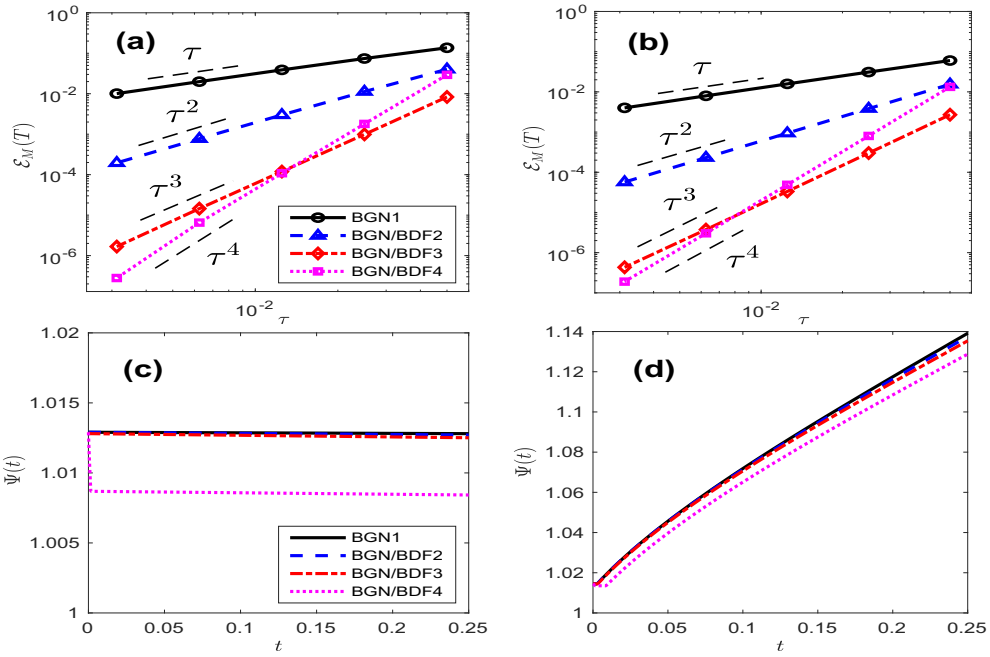

FIG. 1. *Log-log plot of the manifold distance errors via the classical BGN*1 *scheme and the BGN/BDFk schemes* ($2 \le k \le 4$) *at time* $T = 0.25$ *for solving CSF with two various initial curves:* (a) *unit circle and* (b) *ellipse; and the evolution of the mesh distribution function* $\Psi(t)$*:* (c) *unit circle and* (d) *ellipse, where* $N = 640$ *and* $\tau = 1/1280$.

will maintain its quality throughout the evolution. This characteristic significantly enhances the robustness of our BGN/BDFk schemes.

To further investigate the impact of the initial parametrization on the high-order accuracy and mesh quality of the BGN/BDFk algorithms, we conduct additional experiments using the unit circle as the initial curve. We consider two choices for approximating the N-polygon:

- Regular polygon: the nodes are set as

$$\mathbf{X}_i = (\cos(2\pi i/N), \sin(2\pi i/N)), \quad i = 1, \ldots, N.$$

  This ensures an initial mesh ratio of 1 up to machine precision.
- Irregular polygon: the nodes are chosen as

$$\mathbf{X}_i = (\cos(2\pi i/N + 0.1\sin(2\pi i/N)), \sin(2\pi i/N + 0.1\sin(2\pi i/N))),$$
$$i = 1, \ldots, N.$$

  This type of distribution can be traced back to [8, section 3.1]. It is clear that the initial mesh ratio deviates from 1.

Figures 2(a)–2(b) demonstrate the robust temporal convergence of the BGN/BDFk algorithms of kth-order for both initial distributions. Additionally, Figure 2(c) shows that when using a uniform initial distribution, the mesh ratio remains close to 1 up to machine precision. On the other hand, Figure 2(d) illustrates that the BGN/BDF4 algorithm significantly reduces the mesh ratio at the first time step. This is because we compute the polygon Γ^1 using the BGN1 scheme for $\tau/\widetilde{\tau}$ steps (cf. Algorithm 3.3), and the BGN1 scheme demonstrates favorable properties in terms of mesh distribution for curve evolution [8, 10, 28].

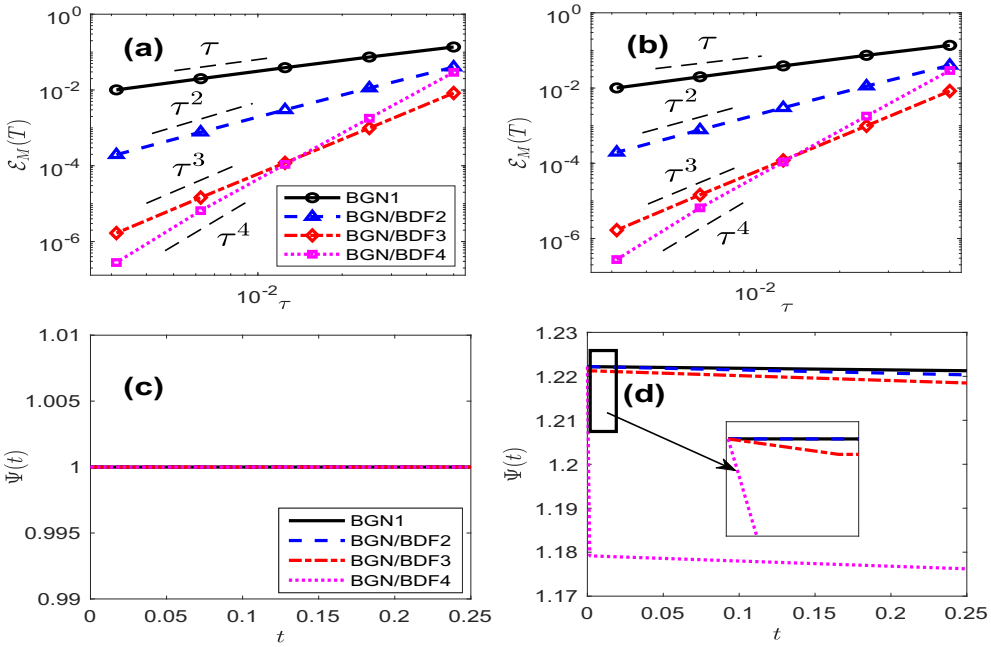

FIG. 2. *Log-log plot of the manifold distance errors at time $T = 0.25$ for solving CSF with a unit circle being the initial curve:* (a) *uniform initial distribution;* (b) *nonuniform initial distribution; and the evolution of the mesh distribution function $\Psi(t)$:* (c) *uniform initial distribution and* (d) *nonuniform initial distribution, where $N = 640$ and $\tau = 1/1280$.*

Example 4.2 (extension to AP-CSF). We extend the BGN/BDFk schemes to the AP-CSF and check the convergence rate of the BGN1 scheme and BGN/BDFk schemes. The initial curve is chosen as an ellipse. The parameters are chosen as $N = 10000$, $T = 0.25$ and 1.

We briefly show how to construct BGN/BDFk schemes for AP-CSF. We first write the governing equation of AP-CSF into the following coupled equations:

$$\begin{aligned} \partial_t \mathbf{X} \cdot \mathbf{n} &= -\kappa + \langle \kappa \rangle, \\ \kappa \mathbf{n} &= -\partial_{ss} \mathbf{X}. \end{aligned} \tag{4.6}$$

The corresponding BGN1 scheme and BGN/BDFk schemes adjust the first equation in (2.2) and (3.1) to their nonlocal versions as

$$\begin{aligned} \left(\frac{\mathbf{X}^{m+1} - \mathbf{X}^m}{\tau}, \varphi^h \mathbf{n}^m \right)^h_{\Gamma^m} &= -\left(\kappa^{m+1} - \langle \kappa^{m+1} \rangle_{\Gamma^m}, \varphi^h \right)^h_{\Gamma^m}, \\ \left(\frac{a\mathbf{X}^{m+1} - \widehat{\mathbf{X}}^m}{\tau}, \varphi^h \widetilde{\mathbf{n}}^{m+1} \right)^h_{\widetilde{\Gamma}^{m+1}} &= -\left(\kappa^{m+1} - \langle \kappa^{m+1} \rangle_{\widetilde{\Gamma}^{m+1}}, \varphi^h \right)^h_{\widetilde{\Gamma}^{m+1}}, \end{aligned}$$

respectively, where $\langle \kappa^{m+1} \rangle_{\Gamma^m} := \frac{(\kappa^{m+1}, 1)^h_{\Gamma^m}}{(1,1)^h_{\Gamma^m}}$, $\langle \kappa^{m+1} \rangle_{\widetilde{\Gamma}^{m+1}} := \frac{(\kappa^{m+1}, 1)^h_{\widetilde{\Gamma}^{m+1}}}{(1,1)^h_{\widetilde{\Gamma}^{m+1}}}$, and the $a, \widehat{\mathbf{X}}^m$ are defined by (3.2)–(3.4).

For the test of temporal convergence rate, we take the ellipse as an example and similarly compute the reference solution using a fine mesh size of $N = 10000$ and

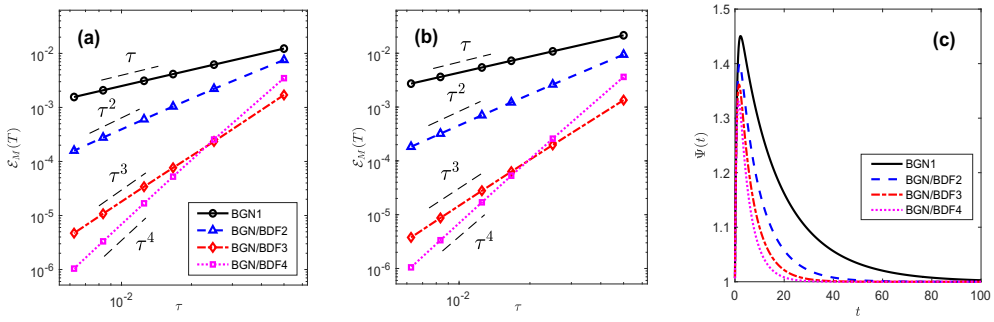

FIG. 3. *Log-log plot of the manifold distance errors for solving the AP-CSF with an ellipse being its initial shape at two different times:* (a) $T = 0.25$, (b) $T = 1$. (c) *The corresponding evolution of the mesh distributional function* $\Psi(t)$, *where* $N = 640$ *and* $\tau = 1/1280$.

a time step of $\tau = 1/1280$. Figures 3(a)–3(b) show the comparison of the temporal convergence rate of the BGN1 scheme and BGN/BDFk schemes at different times $T = 0.25, 1$. It can be clearly observed that the numerical error of BGN/BDFk schemes converge in kth-order, while the classical BGN1 scheme converges only linearly. The evolution of the mesh distribution function is depicted in Figure 3(c) for different schemes with the same computational parameters $N = 640$ and $\tau = 1/1280$. It is worth noting that for long-time simulation, the mesh distribution function $\Psi(t)$ approaches 1, indicating the long-time asymptotic mesh equidistribution, for all kinds of algorithms. Furthermore, we also observe that the high-order algorithms will achieve equidistributed mesh faster and the mesh distribution function is decreasing with respect to the order k, showing the advantage of using high-order BGN/BDFk schemes in achieving a more evenly distributed mesh.

Remark 4.3. In section 3, we highlight the limitations of standard extrapolation approximations for the prediction polygon $\widetilde{\Gamma}^{m+1}$. To illustrate this point, we use the BGN/BDF3 algorithm to solve the AP-CSF with a "flower" initial curve, i.e., a nonconvex curve parametrized by

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

Figure 4 (bottom row) demonstrates that when approximating the integration polygon $\widetilde{\Gamma}^{m+1}$ by extrapolation formula (3.7), the evolution becomes unstable even at very early stage $t = 0.05$, eventually leading to a breakdown of the algorithm (see Figure 4(d2)).

We further compare the evolution of geometric quantities in Figure 5. Figures 5(a1)–5(b1) demonstrate that our BGN/BDF3 scheme effectively preserves the perimeter-decreasing property of AP-CSF and the error of the area is very small. Most importantly, Figure 5(c1) illustrates the long-time asymptotic mesh equidistribution property of our BGN/BDF3 algorithm. In comparison, when approximating the integration polygon by the extrapolation formula, we observe from Figure 5 (bottom row) that the mesh ratio becomes extremely large after some time, ultimately leading to the instability of the BGN/BDF3 algorithm. This highlights the essential role of our treatment for approximations of the prediction polygon $\widetilde{\Gamma}^{m+1}$ through the classical BGN1 scheme or lower-order BGN/BDFk schemes, which ensures the numerical stability and long-time mesh equidistribution.

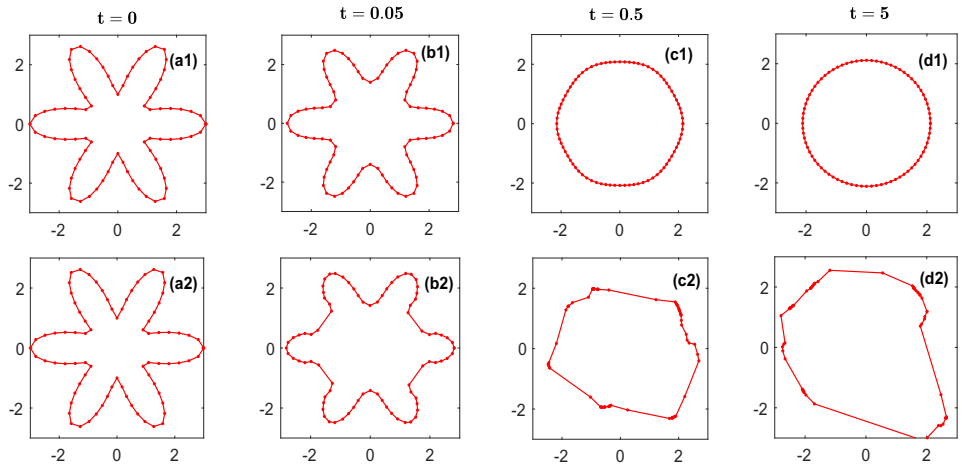

FIG. 4. *Evolution of AP-CSF for the flower initial curve by using the BGN/BDF*3 *algorithm with various choices of predictions* $\widetilde{\mathbf{X}}^{m+1}$. *Top row: Algorithm* 3.2, *i.e., approximating the prediction polygon by a lower-order BGN/BDF*2 *scheme; Bottom row: approximating the prediction polygon by using extrapolation formula* (3.7). *The parameters are chosen as* $N = 80$ *and* $\tau = 1/160$.

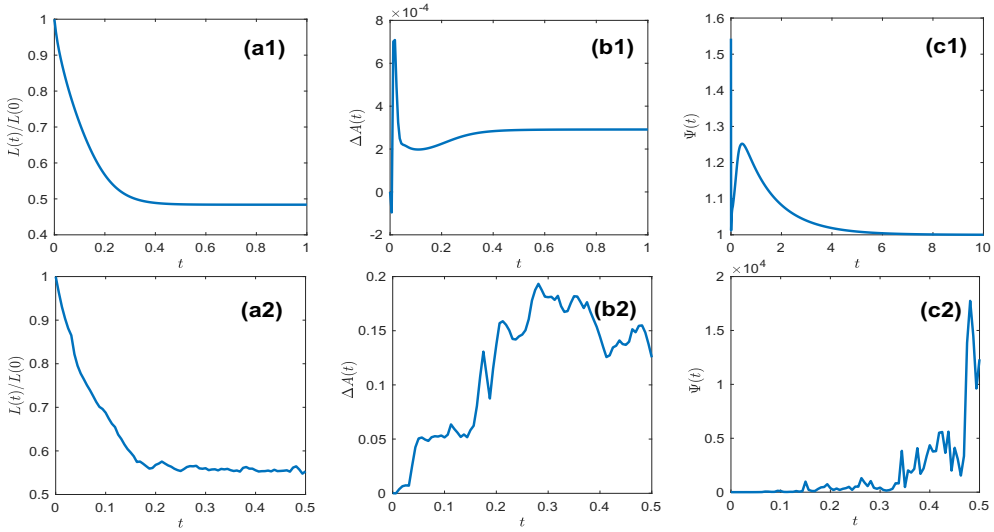

FIG. 5. *Evolution of geometric quantities of AP-CSF for a flower initial shape by using BGN/BDF*3 *algorithm:* (a1)–(a2) *the normalized perimeter;* (b1)–(b2) *the normalized area loss;* (c1)–(c2) *the mesh distribution function* $\Psi(t)$. *Top row: the prediction polygon* $\widetilde{\mathbf{X}}^{m+1}$ *is obtained by a lower-order BGN/BDF*2 *scheme. Bottom row:* $\widetilde{\mathbf{X}}^{m+1}$ *is obtained by the extrapolation formula* (3.7). *The parameters are chosen as* $N = 80$ *and* $\tau = 1/160$.

Example 4.4 (extension to G-MCF). We extend the BGN/BDFk schemes to G-MCF. For the convergence rate test, the initial shape is chosen as a unit circle, the parameters are chosen as $N = 5000$, $T = 0.25$.

The construction of BGN/BDFk schemes for G-MCF is similar to the AP-CSF case. We first rewrite the coupled equations as

$$\begin{aligned} \partial_t \mathbf{X} \cdot \mathbf{n} &= -\beta \kappa^{\alpha}, \\ \kappa \mathbf{n} &= -\partial_{ss} \mathbf{X}. \end{aligned} \tag{4.7}$$

The corresponding BGN1 scheme and BGN/BDFk schemes adjust the first equation in (2.2) and (3.1) to implicit terms as

$$\left(\frac{\mathbf{X}^{m+1}-\mathbf{X}^m}{\tau}, \varphi^h \mathbf{n}^m\right)^h_{\Gamma^m} + \left(\beta(\kappa^{m+1})^{\alpha}, \varphi^h\right)^h_{\Gamma^m} = 0,$$
$$\left(\frac{a\mathbf{X}^{m+1}-\widehat{\mathbf{X}}^m}{\tau}, \varphi^h \widetilde{\mathbf{n}}^{m+1}\right)^h_{\widetilde{\Gamma}^{m+1}} + \left(\beta(\kappa^{m+1})^{\alpha}, \varphi^h\right)^h_{\widetilde{\Gamma}^{m+1}} = 0,$$

respectively, where the $a, \widehat{\mathbf{X}}^m$ are defined similarly. All of the above schemes can be efficiently solved using Newton's iteration method [8, 37].

In order to test the temporal convergence rate for various choices of α and β, we use a unit circle as the initial data. The true solution is given by

$$\mathbf{X}_{\text{true}}(\rho,t) = \begin{cases} (1-(\alpha+1)t)^{\frac{1}{\alpha+1}}(\cos(2\pi\rho), \sin(2\pi\rho)), & \beta = 1, \quad 0 < \alpha \neq 1, \\ e^t(\cos(2\pi\rho), \sin(2\pi\rho)), & \beta = -1, \quad \alpha = -1, \end{cases}$$

where $\rho \in \mathbb{I}$ and $t \in [0, T]$. We use a sufficiently large number of grid points $N = 5000$, and we observe the expected convergence order for different cases, as depicted in Figures 6(a)–6(c). Moreover, Figures 6(d)–6(f) demonstrate that the BGN/BDFk algorithms maintain the good mesh quality comparable to the classical BGN1 scheme, regardless of the specific settings of (α, β).

Example 4.5 (extension to WF). We extend the BGN/BDFk schemes to the fourth-order WF. The initial shape is chosen as a unit circle which yields the exact solution [11]

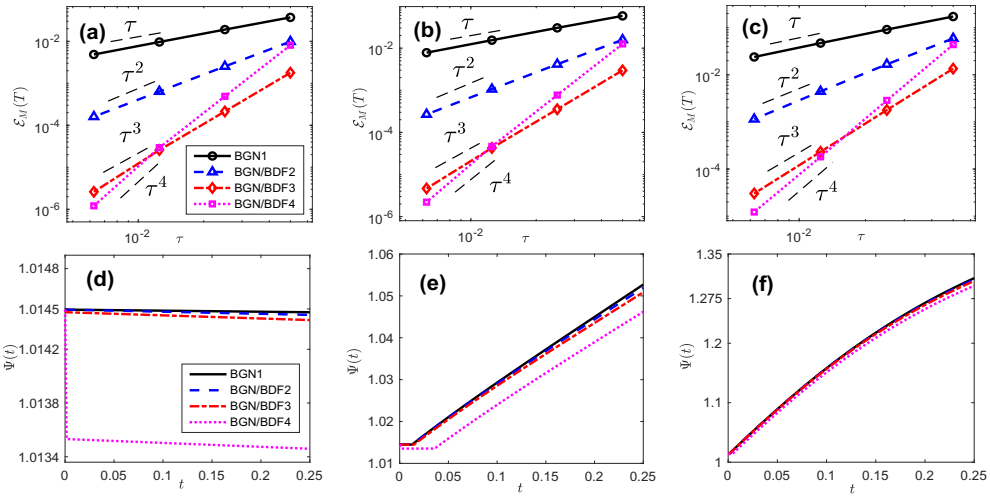

FIG. 6. *(Top row) Log-log plot of the manifold distance errors for solving G-MCF with a unit circle as its initial shape at time* $T = 0.25$ *under different choices of* (α, β)*:* (a) $\beta = 1$, $\alpha = 1/3$, (b) $\beta = 1$, $\alpha = 1/2$, (c) $\beta = -1$, $\alpha = -1$. *(Bottom row) the evolution of the corresponding mesh distributional function* $\Psi(t)$ *under different choices of* (α, β)*:* (d) $\beta = 1$, $\alpha = 1/3$, (e) $\beta = 1$, $\alpha = 1/2$, (f) $\beta = -1$, $\alpha = -1$, *where* $N = 640$ *and* $\tau = 1/1280$.

$$\mathbf{X}_{\text{true}}(\rho,t)=(1+2t)^{\frac{1}{4}}(\cos(2\pi\rho),\sin(2\pi\rho)),\quad \rho\in\mathbb{I},\quad t>0.$$

Similar to the AP-CSF and G-MCF cases, we first reformulate the coupled equations as

$$\begin{aligned}\partial_t\mathbf{X}\cdot\mathbf{n}&=\partial_{ss}\kappa-\frac{1}{2}\kappa^3,\\ \kappa\mathbf{n}&=-\partial_{ss}\mathbf{X}.\end{aligned}\tag{4.8}$$

The corresponding BGN1 scheme [11] and BGN/BDFk schemes adjust the first equation in (2.2) and (3.1) to

$$\left(\frac{\mathbf{X}^{m+1}-\mathbf{X}^m}{\tau},\varphi^h\mathbf{n}^m\right)^h_{\Gamma^m}+\left(\partial_s\kappa^{m+1},\partial_s\varphi^h\right)_{\Gamma^m}=-\frac{1}{2}\left((\kappa^{m+1})^3,\varphi^h\right)^h_{\Gamma^m},$$
$$\left(\frac{a\mathbf{X}^{m+1}-\widehat{\mathbf{X}}^m}{\tau},\varphi^h\widetilde{\mathbf{n}}^{m+1}\right)^h_{\widetilde{\Gamma}^{m+1}}+\left(\partial_s\kappa^{m+1},\partial_s\varphi^h\right)_{\widetilde{\Gamma}^{m+1}}=-\frac{1}{2}\left((\kappa^{m+1})^3,\varphi^h\right)^h_{\widetilde{\Gamma}^{m+1}},$$

respectively.

Figures 7(a)–7(b) display the convergence and the evolution of the mesh ratio $\Psi(t)$, respectively, which suggest that the BGN/BDFk scheme converges at the kth-order in terms of manifold distance and it maintains the good mesh quality throughout the simulation of the fourth-order WF.

4.2. Surface evolution. In this subsection, we mainly consider the following geometric flows of surface evolution in three-dimensional space:

- MCF, which is a second-order geometric flow defined by

$$\mathcal{V}=-\mathcal{H}\,\mathbf{n},\tag{4.9}$$

  where $\mathcal{H}$ is the mean curvature of hypersurface;
- Surface diffusion flow (SDF), which is a fourth-order geometric flow defined by

$$\mathcal{V}=\Delta_\Gamma\mathcal{H}\,\mathbf{n},\tag{4.10}$$

  where Δ_Γ is the surface Laplacian.

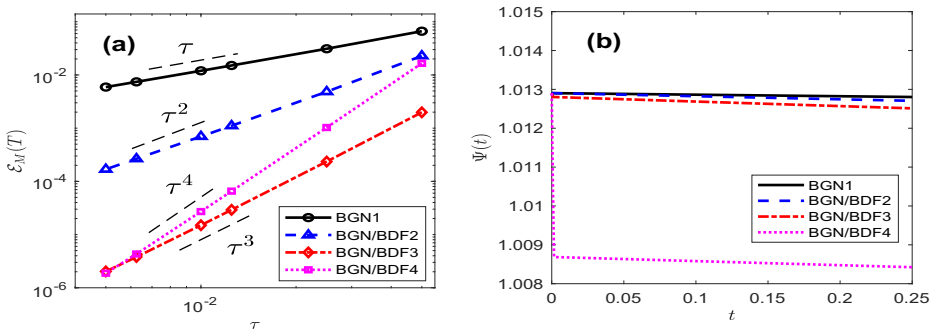

FIG. 7. (a) *Log-log plot of the numerical errors for solving the WF with unit circle initial shape at time* $T=0.25$. (b) *The evolution of the mesh distribution function* $\Psi(t)$, *where the parameters are chosen as* $N=640$ *and* $\tau=1/1280$.

Given an initial surface $\Gamma(0)$, we use the MATLAB package `DistMesh` [38] to give an initial triangulation polyhedron Γ^0 with good mesh quality. It is worth mentioning that the manifold distance (4.4) can be easily extended to the three-dimensional case, and we still use this shape metric to measure the difference between two polyhedrons. We point out that other shape metrics such as Hausdorff distance [6, 28] can also effectively characterize the convergence rates of our schemes. The numerical error and convergence order are similarly defined as (4.5).

We further investigate the evolution of the following geometric quantities: (1) the relative volume loss $\Delta V(t)$, (2) the normalized surface area $S(t)/S(0)$, which are defined, respectively, for $m \geq 0$ as

$$\Delta V(t)|_{t=t_m} = \frac{V^m - V^0}{V^0}, \quad \left.\frac{S(t)}{S(0)}\right|_{t=t_m} = \frac{S^m}{S^0},$$

where V^m is the volume enclosed by the polyhedron determined by $\mathbf{X}^m$, and S^m represents the surface area of the polyhedron. To evaluate the mesh quality of a polyhedron, we introduce two mesh distribution functions $r_h(t)$ and $r_a(t)$ [10, 26] defined as

$$r_h(t)|_{t=t_m} := \frac{\max_j \max\{\|\mathbf{q}^m_{j_1} - \mathbf{q}^m_{j_2}\|, \|\mathbf{q}^m_{j_2} - \mathbf{q}^m_{j_3}\|, \|\mathbf{q}^m_{j_3} - \mathbf{q}^m_{j_1}\|\}}{\min_j \min\{\|\mathbf{q}^m_{j_1} - \mathbf{q}^m_{j_2}\|, \|\mathbf{q}^m_{j_2} - \mathbf{q}^m_{j_3}\|, \|\mathbf{q}^m_{j_3} - \mathbf{q}^m_{j_1}\|\}},$$
$$r_a(t)|_{t=t_m} := \frac{\max_j |\sigma^m_j|}{\min_j |\sigma^m_j|}.$$

Example 4.6 (extension to MCF). The BGN/BDFk schemes can be extended straightforwardly to the MCF in $\mathbb{R}^3$. It suffices to replace the curvature κ^{m+1} in (3.1) by the mean curvature $\mathcal{H}^{m+1}$.

For the test of convergence order, we choose the initial surface as a unit sphere, which remains as a sphere with radius $R(t)$ given by

$$R(t) = \sqrt{1 - 4t}, \quad t \in [0, 1/4).$$

For the initial triangulation, we fix the fine mesh size $(J, K) = (93608, 46806)$, where J and K represent the number of triangles and vertices of polyhedra, respectively. Figure 8(a) shows a log-log plot of the manifold errors at time $T = 0.05$ for the BGN1 scheme and BGN/BDFk schemes with $2 \leq k \leq 3$, which demonstrates the

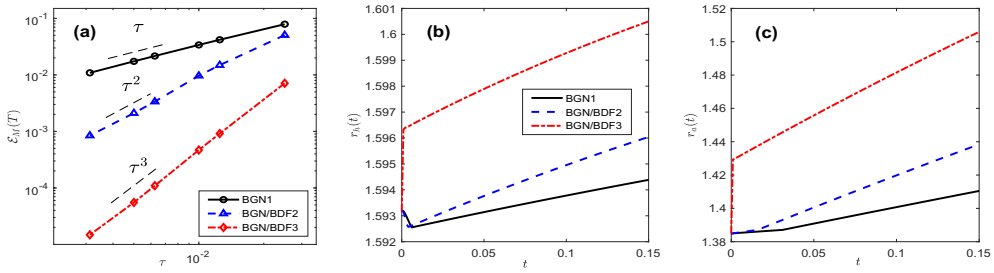

FIG. 8. (a) *Log-log plot of the manifold distance errors for solving the MCF until* $T = 0.05$ *with unit sphere initial shape, where the spatial parameters are chosen as* $(J, K) = (93608, 46806)$. *The evolution of the mesh distribution functions:* (b) $r_h(t)$ *and* (c) $r_a(t)$, *where the parameters are chosen as* $(J, K) = (14888, 7446)$ *and* $\tau = 1/1000$.

expected convergence rate. From Figures 8(b)–8(c), we observe that the two mesh distribution functions $r_h(t)$ and $r_a(t)$ remain below 2 throughout the evolution, indicating the good mesh quality of our proposed BGN/BDFk algorithms. We note that for the BGN/BDF3 algorithm, the two mesh distribution functions increase rapidly at the first step. The reason is that we need a fine time step to initiate BGN/BDF3 scheme to ensure the third-order accuracy (see Algorithm 3.2), however, as reported in the recent paper [18], the classical BGN1 scheme for surface evolution may result in mesh clustering for very fine time step sizes.

For the evolution test, we apply BGN/BDFk schemes to two benchmark dumbbell examples [18, 24]. The initial surface is a dumbbell-shaped surface with a fat waist given by the following parametrization:

$$\mathbf{X}(\theta,\varphi)=\begin{pmatrix}\cos\varphi\\(0.6\cos^2\varphi+0.4)\cos\theta\sin\varphi\\(0.6\cos^2\varphi+0.4)\sin\theta\sin\varphi\end{pmatrix},\quad \theta\in[0,2\pi),\quad \varphi\in[0,\pi], \tag{4.11}$$

or a dumbbell shape with a thin waist parametrized by

$$\mathbf{X}(\theta,\varphi)=\begin{pmatrix}\cos\varphi\\(0.7\cos^2\varphi+0.3)\cos\theta\sin\varphi\\(0.7\cos^2\varphi+0.3)\sin\theta\sin\varphi\end{pmatrix},\quad \theta\in[0,2\pi),\quad \varphi\in[0,\pi]. \tag{4.12}$$

The numerical simulations are presented in Figures 9 and 10, respectively. We can clearly observe that MCF evolves the first dumbbell shape to a round point (cf. Figures 9(d1), 9(d2), and 9(d3)), and develops a neck pinch singularity for the second dumbbell shape in finite time (cf. Figure 10(f)). We observe that the BGN/BDFk algorithms achieve a similar mesh quality to the classical BGN1 scheme. The mesh is slightly distorted in Figure 9(d3) since we use a fine time step size to predict the polyhedron at the first step (cf. Algorithm 3.2), which may result in mesh distortion for surface evolution as reported in [18]. Nevertheless, we have successfully obtained the blowup times, that is $t=0.0908$ for the first dumbbell shape and $t=0.0591$ for the

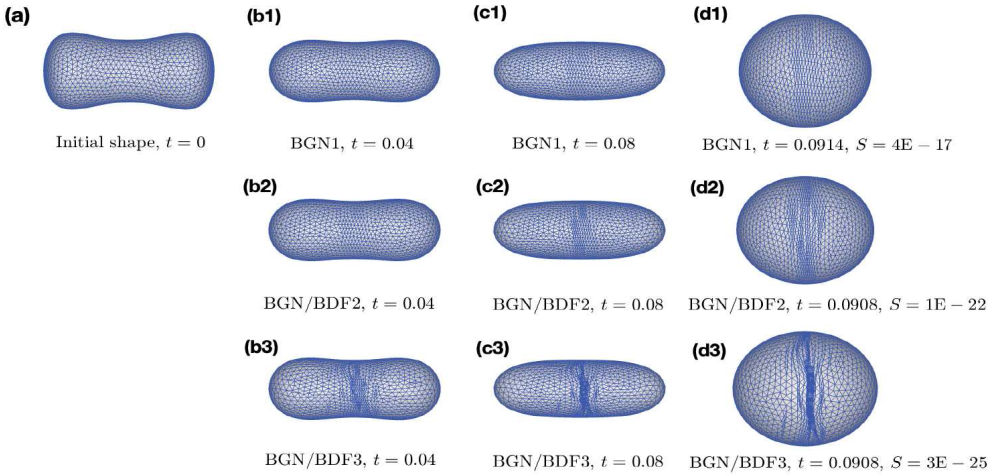

FIG. 9. *Evolution of MCF by the BGN*1 *scheme (first row), BGN/BDF*2 *scheme (second row), and BGN/BDF*3 *scheme (third row) starting from the first dumbbell with a fat waist (note that the images are scaled). The initial surface is triangulated into* 3604 *triangles with* 1804 *vertices and the time step* $\tau=1/10000$.

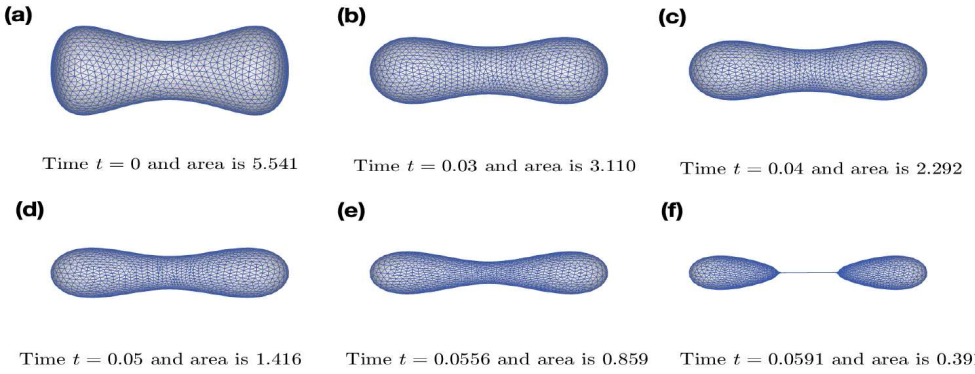

FIG. 10. *Evolution of MCF by the BGN/BDF*2 *scheme starting from the first dumbbell with a thin waist (the images are scaled). The initial surface is triangulated into* 3276 *triangles with* 1640 *vertices and the time step* $\tau = 1/10000$.

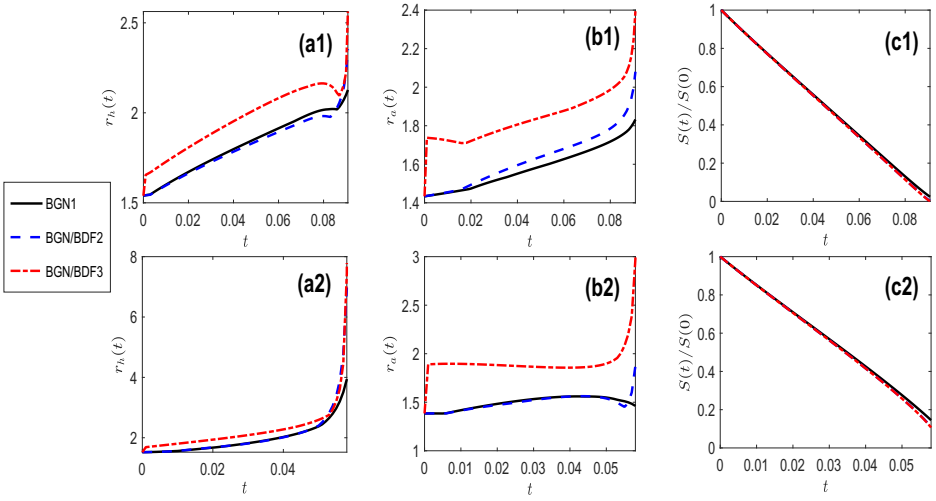

FIG. 11. *Evolution of geometric quantities in MCF for two initial dumbbell shapes by using BGN/BDFk algorithms: the first dumbbell example* (4.11) *(top row); the second dumbbell example* (4.12) *(bottom row). The mesh distribution functions* $r_h(t)$*:* (a1)–(a2)*; and* $r_a(t)$*:* (b1)–(b2)*. The normalized surface area:* (c1)–(c2)*. The time step is chosen as* $\tau = 1/1000$.

second dumbbell shape, respectively. These results are comparable to those reported in [24, sections 7.2 and 7.3] and [18, Example 3.2].

The evolution of the two mesh distribution functions $r_h(t)$, $r_a(t)$, and the normalized surface area are plotted in Figure 11. As clearly shown in Figures 11(a)–(b), our high-order schemes exhibit similar mesh behavior to the classical BGN1 scheme before the blowup time. Moreover, Figures 11(c1)–11(c2) demonstrate that our methods preserve the geometric property (i.e., the decreasing surface area) of MCF very well.

Example 4.7 (extension to SDF). We investigate the performance of BGN/BDFk schemes when applied to SDF in this example.

Specifically, the schemes for SDF can be derived similarly to the MCF case and it suffices to adjust the first equation as

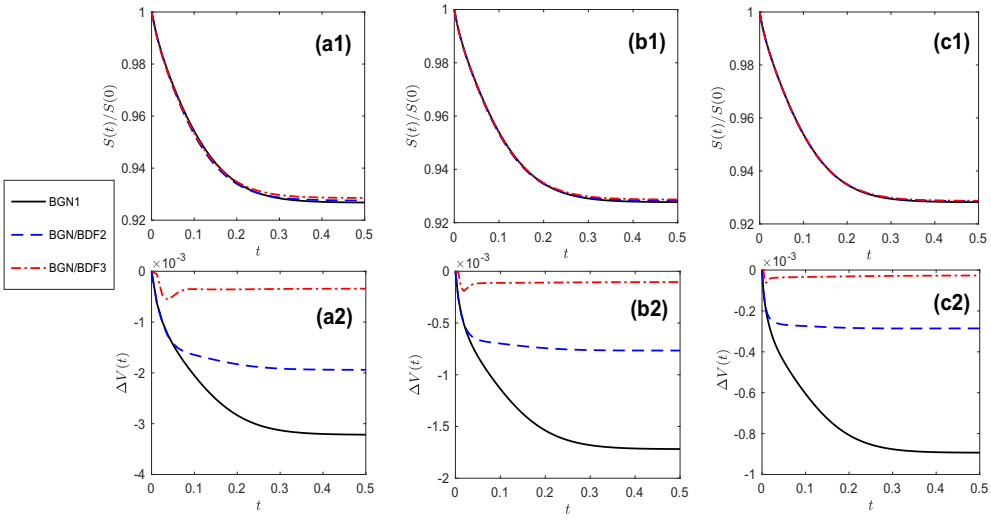

FIG. 12. *Evolution of geometric quantities for SDF applied to an initial ellipsoid using BGN/BDFk algorithms. Top row: the normalized surface area; bottom row: the relative volume loss. The triangulation parameters are set as* $(J,K) = (24952, 12478)$ *and the time step is set as* $\tau = 1/160$ *for* (a1), (a2)*;* $\tau = 1/320$ *for* (b1), (b2)*; and* $\tau = 1/640$ *for* (c1), (c2).

$$\left(\frac{\mathbf{X}^{m+1}-\mathbf{X}^m}{\tau}, \varphi^h \mathbf{n}^m\right)^h_{\Gamma^m} + \left(\nabla_{\Gamma^m}\mathcal{H}^{m+1}, \nabla_{\Gamma^m}\varphi^h\right)_{\Gamma^m} = 0,$$

$$\left(\frac{a\mathbf{X}^{m+1}-\widehat{\mathbf{X}}^m}{\tau}, \varphi^h \widetilde{\mathbf{n}}^{m+1}\right)^h_{\widetilde{\Gamma}^{m+1}} + \left(\nabla_{\Gamma^m}\mathcal{H}^{m+1}, \nabla_{\Gamma^m}\varphi^h\right)_{\widetilde{\Gamma}^{m+1}} = 0.$$

Instead of conducting a convergence order test, we demonstrate the superiority of our high-order schemes by examining the relative volume loss $\Delta V(t)$. The initial surface is a $2:1:1$ ellipsoid defined by the equation $x^2/4 + y^2 + z^2 = 1$, and we set the triangulation parameters as $(J,K) = (24952, 12478)$. Figure 12 illustrates that all methods maintain the geometric properties of SDF, i.e., the decrease of the surface area and the conservation of the volume enclosed by the surface. Additionally, Figures 12(a2), 12(b2), and 12(c2) highlight that our high-order algorithms result in significantly smaller volume loss compared to BGN1 scheme [10], thus showcasing the accuracy of our proposed methods.

In Figure 13, we present several evolution snapshots of the $2:1:1$ ellipsoid towards its equilibrium using the BGN/BDF2 scheme with the surface triangulation parameters $(J,K) = (2780, 1392)$. Furthermore, Figure 14 depicts the comparison of the two mesh distribution functions $r_h(t)$ and $r_a(t)$ for both the BGN1 and BGN/BDFk schemes. It is worth noting that our high-order method successfully evolves the ellipsoid into a perfect sphere while maintaining good mesh quality, similarly to the classical BGN scheme.

In the last experiment, we apply our BGN/BDFk schemes to conduct the evolution of an $8 \times 1 \times 1$ cuboid driven by SDF. As a benchmark example, this cuboid pinches off at finite time [6, 10]. We select the triangulation parameters as $(J,K) = (2600, 1302)$, and the time step is fixed as $\tau = 1/2500$. The evolution is depicted in Figure 15, where we observe that the pinch-off event occurs at time $t = 0.3636$ for the BGN1 scheme, $t = 0.3692$ for the BGN/BDF2 algorithm, and $t = 0.3704$ for the BGN/BDF3 algorithm, respectively. The pinch-off time of this benchmark cuboid

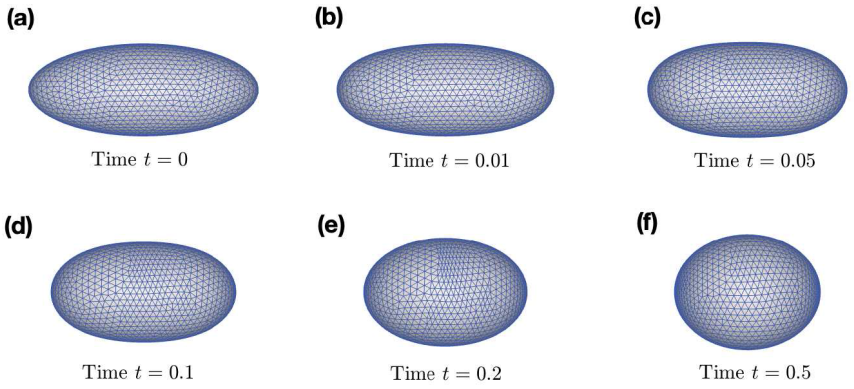

FIG. 13. *Several evolution snapshots of SDF by the BGN/BDF2 scheme, where the initial shape is chosen as the* $2:1:1$ *ellipsoid, and the triangulation parameters* $(J,K) = (2780, 1392)$ *and* $\tau = 10^{-4}$.

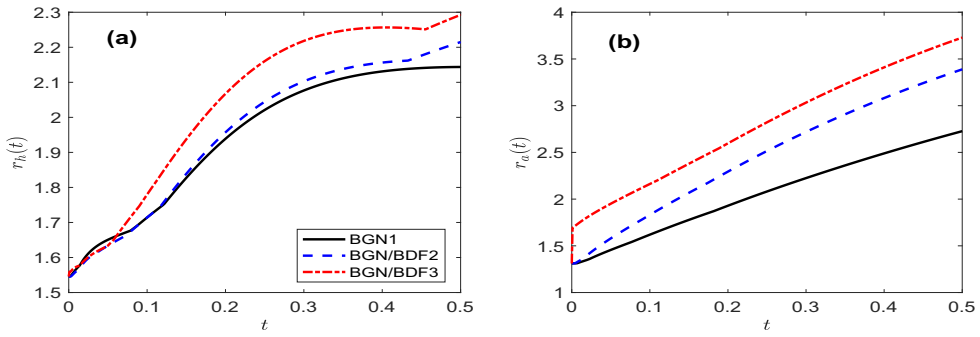

FIG. 14. *Evolution of mesh distribution functions of SDF for the* $2:1:1$ *ellipsoid:* (a) $r_h(t)$, (b) $r_a(t)$, *where* $(J,K) = (2780, 1392)$ *and* $\tau = 10^{-4}$.

example was reported as $t = 0.369$ in [10, section 5.5] using a tedious time adaptive method, and it was predicted as $t = 0.370$ in [6, section 4.2] by using an implicit method. We emphasize that our predicted pinch-off time agrees very well with the previous results [6, 10], while requiring one to solve only two or three linear systems at each time step. Furthermore, the monotonic increasing behavior of pinch-off times in Figures 15(d1)–15(d3) indicates that the high-order schemes provide a better prediction of the pinch-off time.

5. Conclusion. We have proposed a type of novel temporal high-order (second-order to fourth-order), parametric finite element method based on the BGN formulation [7, 8, 12] for solving different types of geometric flows of curves and surfaces, including CSF, AP-CSF, G-MCF, WF, MCF, and SDF. Our approach is constructed based on the BGN formulation [7, 8, 12], the BDF in time, and linear finite element approximation in space. We carefully choose the prediction polygon $\widetilde{\Gamma}^{m+1}$ to ensure that the approximation errors of all quantities are at $\mathcal{O}(\tau^k)$. The key to the success of BGN/BDFk schemes is that $\widetilde{\Gamma}^{m+1}$ should be given by solving lower-order BGN/BDFk schemes, instead of standard extrapolation, to maintain the mesh quality, which is very essential for the simulation of geometric flows. Extensive numerical experiments demonstrate the expected, high-order accuracy and improved performance compared to the classical BGN scheme.

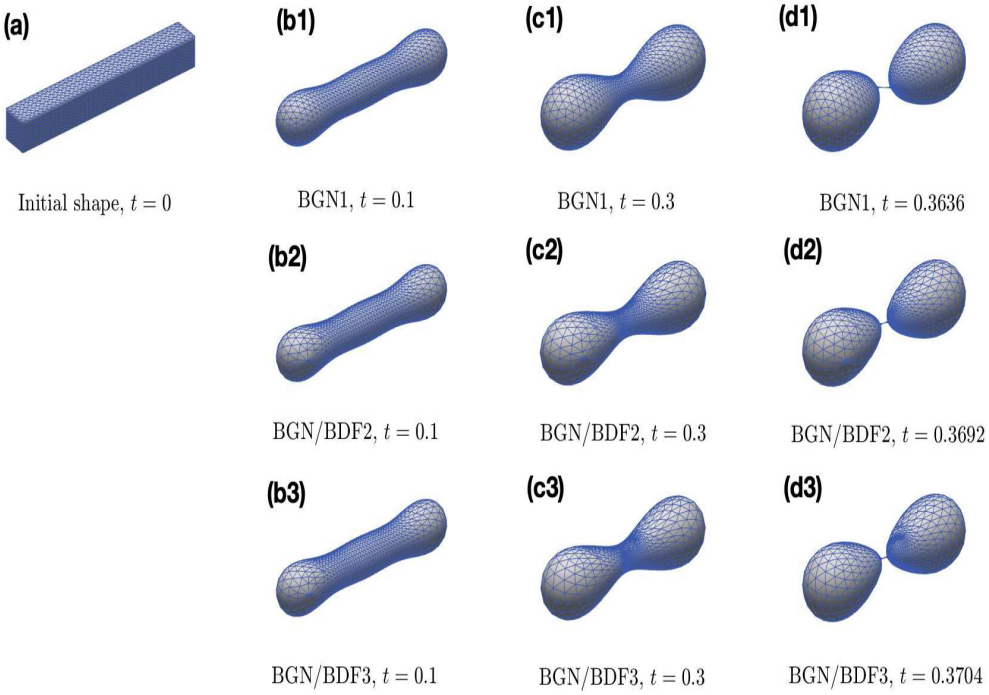

FIG. 15. *Investigation of pinch-off time of an* $8 \times 1 \times 1$ *cuboid driven by SDF by using the BGN*1 *scheme (first row), BGN/BDF*2 *algorithm (second row) and BGN/BDF*3 *algorithm (third row), where* $(J, K) = (2600, 1302)$ *and* $\tau = 1/2500$.

However, designing high-order schemes that preserve the geometric structure, specifically reducing the perimeter and conserving the enclosed area at the discrete level, remains challenging for certain geometric flows such as AP-CSF and SDF [29], which is our future work.

Acknowledgments. The numerical calculations in this paper were partially performed on the supercomputing system in the Supercomputing Center of Wuhan University.

REFERENCES

[1] G. BAI AND B. LI, *A new approach to the analysis of parametric finite element approximations to mean curvature flow*, Found. Comput. Math., to appear, https://doi.org/10.1007/s10208-023-09622-x.

[2] M. BALAŽOVJECH AND K. MIKULA, *A higher order scheme for a tangentially stabilized plane curve shortening flow with a driving force*, SIAM J. Sci. Comput., 33 (2011), pp. 2277–2294.

[3] E. BÄNSCH, P. MORIN, AND R. H. NOCHETTO, *A finite element method for surface diffusion: The parametric case*, J. Comput. Phys., 203 (2005), pp. 321–343.

[4] W. BAO, W. JIANG, AND Y. LI, *A symmetrized parametric finite element method for anisotropic surface diffusion of closed curves*, SIAM J. Numer. Anal., 61 (2023), pp. 617–641.

[5] W. BAO, W. JIANG, Y. WANG, AND Q. ZHAO, *A parametric finite element method for solid-state dewetting problem with anisotropic surface energies*, J. Comput. Phys., 330 (2017), pp. 380–400.

[6] W. BAO AND Q. ZHAO, *A structure-preserving parametric finite element method for surface diffusion*, SIAM J. Numer. Anal., 59 (2021), pp. 2775–2799.

[7] J. W. Barrett, H. Garcke, and R. Nürnberg, *A parametric finite element method for fourth order geometric evolution equations*, J. Comput. Phys., 222 (2007), pp. 441–467.
[8] J. W. Barrett, H. Garcke, and R. Nürnberg, *On the variational approximation of combined second and fourth order geometric evolution equations*, SIAM J. Sci. Comput., 29 (2007), pp. 1006–1041.
[9] J. W. Barrett, H. Garcke, and R. Nürnberg, *Numerical approximation of anisotropic geometric evolution equations in the plane*, IMA J. Numer. Anal., 28 (2008), pp. 292–330.
[10] J. W. Barrett, H. Garcke, and R. Nürnberg, *On the parametric finite element approximation of evolving hypersurfaces in* $\mathbb{R}^3$, J. Comput. Phys., 227 (2008), pp. 4281–4307.
[11] J. W. Barrett, H. Garcke, and R. Nürnberg, *Parametric approximation of Willmore flow and related geometric evolution equations*, SIAM J. Sci. Comput., 31 (2008), pp. 225–253.
[12] J. W. Barrett, H. Garcke, and R. Nürnberg, *Parametric finite element method approximations of curvature driven interface evolutions*, in Geometric Partial Differential Equations. Part I, Handb. Numer. Anal. 21, North-Holland, Amsterdam, 2020, pp. 275–423.
[13] T. Binz and B. Kovács, *A convergent finite element algorithm for generalized mean curvature flows of closed surfaces*, IMA J. Numer. Anal., 42 (2022), pp. 2545–2588.
[14] F. Cao, *Geometric Curve Evolution and Image Processing*, Lecture Notes in Math. 1805, Springer, Berlin, 2003.
[15] K. Deckelnick and G. Dziuk, *On the approximation of the curve shortening flow*, in Calculus of Variations, Applications and Computations, Pont-à-Mousson, 1994, Pitman Res. Notes Math. Ser. 326, Longman Sci. Tech., Harlow, UK, 1995, pp. 100–108.
[16] K. Deckelnick, G. Dziuk, and C. M. Elliott, *Computation of geometric partial differential equations and mean curvature flow*, Acta Numer., 14 (2005), pp. 139–232.
[17] Q. Du, C. Liu, R. Ryham, and X. Wang, *A phase field formulation of the Willmore problem*, Nonlinearity, 18 (2023), pp. 1249–1267.
[18] B. Duan and B. Li, *New artificial tangential motions for parametric finite element approximation of surface evolution*, SIAM J. Sci. Comput., 46 (2024), pp. A587–A608.
[19] G. Dziuk, *An algorithm for evolutionary surfaces*, Numer. Math., 58 (1990), pp. 603–611.
[20] G. Dziuk, *Convergence of a semi-discrete scheme for the curve shortening flow*, Math. Models Methods Appl. Sci., 4 (1994), pp. 589–606.
[21] G. Dziuk, *Discrete anisotropic curve shortening flow*, SIAM J. Numer. Anal., 36 (1999), pp. 1808–1830.
[22] G. Dziuk, *Computational parametric Willmore flow*, Numer. Math., 111 (2008), pp. 55–80.
[23] G. Dziuk, E. Kuwert, and R. Schatzle, *Evolution of elastic curves in* $\mathbb{R}^n$*: Existence and computation*, SIAM J. Math. Anal., 33 (2002), pp. 1228–1245.
[24] C. M. Elliott and 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), pp. 543–603.
[25] S. Esedoglu, S. J. Ruuth, and R. Tsai, *Threshold dynamics for high order geometric motions*, Interfaces Free Bound., 10 (2008), pp. 263–282.
[26] J. Hu and B. Li, *Evolving finite element methods with an artificial tangential velocity for mean curvature flow and Willmore flow*, Numer. Math., 152 (2022), pp. 127–181.
[27] W. Jiang and B. Li, *A perimeter-decreasing and area-conserving algorithm for surface diffusion flow of curves*, J. Comput. Phys., 443 (2021), 110531.
[28] W. Jiang, C. Su, and G. Zhang, *A second-order in time, BGN-based parametric finite element method for geometric flows of curves*, J. Comput. Phys., 514 (2024), 113220.
[29] W. Jiang, C. Su, and 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), pp. 1989–2010.
[30] B. Kovács, B. Li, and C. Lubich, *A convergent evolving finite element algorithm for mean curvature flow of closed surfaces*, Numer. Math., 143 (2019), pp. 797–853.
[31] B. Kovács, B. Li, and C. Lubich, *A convergent algorithm for forced mean curvature flow driven by diffusion on the surface*, Interfaces Free Bound., 22 (2020), pp. 443–464.
[32] B. Kovács, B. Li, and C. Lubich, *A convergent evolving finite element algorithm for Willmore flow of closed surfaces*, Numer. Math., 149 (2021), pp. 595–643.
[33] R. J. Leveque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*, SIAM, Philadelphia, 2007.
[34] J. A. Mackenzie, M. Nolan, C. F. Rowlatt, and R. H. Insall, *An adaptive moving mesh method for forced curve shortening flow*, SIAM J. Sci. Comput., 41 (2019), pp. A1170–A1200.
[35] K. Mikula and D. Ševčovič, *A direct method for solving an anisotropic mean curvature flow of plane curves with an external force*, Math. Methods Appl. Sci., 27 (2004), pp. 1545–1565.

[36] W. W. Mullins, *Two-dimensional motion of idealized grain boundaries*, J. Appl. Phys., 27 (1956), pp. 900–904.
[37] L. Pei and Y. Li, *A structure-preserving parametric finite element method for area-conserved generalized mean curvature flow*, J. Sci. Comput., 96 (2023), 6.
[38] P. O. Persson, *Mesh Generation for Implicit Geometries*, Ph.D. thesis, Department of Mathematics, MIT, Cambridge, MA, 2004.
[39] S. J. Ruuth, B. Merriman, and S. Osher, *Convolution-generated motion as a link between cellular automata and continuum pattern dynamics*, J. Comput. Phys., 151 (2004), pp. 836–861.
[40] G. Sapiro, *Geometric Partial Differential Equations and Image Analysis*, Cambridge University Press, Cambridge, 2001.
[41] D. Ševčovivč and K. Mikula, *Evolution of plane curves driven by a nonlinear function of curvature and anisotropy*, SIAM J. Appl. Math., 61 (2001), pp. 1473–1501.
[42] J. Steinhilber, *Numerical Analysis for Harmonic Maps Between Hypersurfaces and Grid Improvement for Computational Parametric Geometric Flows*. Ph.D. thesis, Albert-Ludwigs-Universität Freiburg, Freiburg, Germany, 2014.
[43] Q. Zhao, W. Jiang, and W. Bao, *A parametric finite element method for solid-state dewetting problems in three dimensions*, SIAM J. Sci. Comput., 42 (2020), pp. B327–B352.
[44] Q. Zhao, W. Jiang, and W. Bao, *An energy-stable parametric finite element method for simulating solid-state dewetting*, IMA J. Numer. Anal., 41 (2021), pp. 2026–2055.