SIAM J. SCI. COMPUT.
Vol. 42, No. 5, pp. A3030–A3054
© 2020 Society for Industrial and Applied Mathematics

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

SPECTRALLY ACCURATE ALGORITHM FOR POINTS REDISTRIBUTION ON CLOSED CURVES*

YUNCHANG SEOL† AND MING-CHIH LAI‡

Abstract. In this paper, we present a novel numerical method that redistributes unevenly given points on an evolving closed curve to satisfy equi-arclength(-like) condition. Without substantial difficulty, it is also capable of remeshing or employing adaptive mesh refinement. The key idea is to find the discrete inverse of the arclength(-like) function in the framework of the Fourier spectral method to obtain overall spectral accuracy. Both equi-arclength and curvature-dependent redistributions are extensively studied, and their spectral accuracy is verified by application to smoothly perturbed points on various curves. We further confirm that our method converges even for the points being perturbed nonsmoothly and randomly. To leverage the robustness of our method, a remeshing technique is applied in which the accuracy is not affected. Application to a periodic planar curve without any modification of our algorithm is also discussed. Then, to show the practical applicability, an evolving curve with large deformation is studied by coupling with point redistribution and remeshing in various flows such as mean curvature flow, Willmore flow, and Stokes flow.

Key words. redistribution, reparametrization, remeshing, equi-arclength, spectral method, monitor function

AMS subject classifications. 65D15, 65D17

DOI. 10.1137/20M1314690

1. Introduction.

In fluid-structure interaction problems with large deformations, it is essential to maintain the mesh quality of a discretized structure boundary. For instance, a droplet in shear flow undergoes a rotating motion or an elongational deformation [22]. In the former case, the points approximating the droplet interface can be clustered, causing numerical instability, and in the latter case, the points can be isolated, causing volume loss. To avoid such difficulties, in this paper, we propose a spectrally accurate numerical method that redistributes the points on an evolving closed curve satisfying desired conditions.

A numerical method, known as de Boor's algorithm described in [7, 25] and [12, Chap. 2.2.1], is a simple but effective approach for finding equi-arclength redistribution of points. This method is based on piecewise constant or linear interpolation of monitor function and still commonly used to generate an initially good approximation of a given curve before applying other methods which establish better approximation [17]. In [25], its convergence rate is observed to be second-order for data given in the Sobolev space $W^{2,\infty}$. To achieve high accuracy, the spline is popularly used in many works [2, App. B] and [23, 24, 1], where the applicability of equi-arclength redistribution is shown to be worthy of study in computer-aided design and fluid dynamics. Grid generation is also closely related [16, 12]. To achieve higher accuracy, Fourier approximation is used particularly in pattern recognition and data fitting [13, 10, 4, 15].

*Submitted to the journal's Methods and Algorithms for Scientific Computing section January 23, 2020; accepted for publication (in revised form) June 22, 2020; published electronically October 1, 2020.

https://doi.org/10.1137/20M1314690

Funding: The work of the second author was partially supported by the Ministry of Science and Technology of Taiwan under research grant MOST-107-2115-M-009-016-MY3 and by NCTS.

†Corresponding author. Department of Applied Mathematics, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu 30010, Taiwan (ycseol@math.nctu.edu.tw, https://sites.google.com/site/yunchangseol/).

‡Department of Applied Mathematics, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu 30010, Taiwan (mclai@math.nctu.edu.tw, http://jupiter.math.nctu.edu.tw/∼mclai/).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

To the best of our knowledge, however, a highly accurate algorithm has not been presented yet, whereas our method indeed achieves spectral accuracy.

Our numerical method proposed here can be considered as an accuracy-improved version of de Boor's algorithm, in which the idea is to find the inverse of an arclength(-like) function in a discrete sense. As discussed in [21], it is rarely possible to satisfy the exact equi-arclength condition since an integration is only numerically approximated. To achieve spectral accuracy in our method, however, most of the variables are represented in Fourier series as possible, and then the spectral calculus is performed extensively. In [4], the authors presented a similar algorithm but based on the function of tangent angle to a closed curve, hence the procedure is a little bit different from de Boor's algorithm. On the other hand, our method is based on the arclength(-like) function as in de Boor's algorithm so that the understanding of our algorithm is transparent, seen as a straightforward extension in the framework of Fourier spectral methods.

In the aspect of tangential redistribution of points, there is another type of method, namely a moving mesh method presented in [11, 14, 12, 17]. Such an approach requires the velocity information with respect to time and space, while our method requires only the position vector of given points at fixed time. Thus, to save computational cost in time-dependent dynamics, our method can be applied at certain selected times when the points are unevenly distributed. Meanwhile, in practical situations, growing complexity of an evolving boundary gives rise to the importance of remeshing or adaptive mesh refinement (AMR). Our method is also capable of controlling the number of points without much effort to modify the code. We need to find solutions associated with only the desired number of points, while the rest of the procedure remains the same.

An outline of this paper is as follows. In section 2, the reparametrization of a closed curve under equi-arclength(-like) conditions is described. In section 3, its numerical scheme is realized by redistributions of points, and therein the equi-arclength and curvature-dependent redistributions are described together with spectral calculus. A remeshing algorithm controlling the number of points is also presented. In section 4, to estimate accuracy and convergence rate, the equi-arclength redistribution is applied to perturbed points on the unit circle. Then in section 5, to identify the qualitative effectiveness, both redistributions are applied to various fixed curves and evolving curves in dynamic flow. Conclusions and future works are given in section 6.

2. Curve reparametrization and related equations. In this section, we begin with an introduction to the fundamentals of differential geometry of a curve, and then describe the equi-arclength(-like) reparametrization of a curve.

2.1. Differential geometry of a curve. For a sufficiently smooth closed curve $\mathbf{X}(\theta) = (X(\theta), Y(\theta))$ with a given parameter θ, the unit tangent vector is

$$\boldsymbol{\tau}(\theta) = \mathbf{X}_\theta / \|\mathbf{X}_\theta\| := (\tau_1, \tau_2),$$

and the outward unit normal vector is

$$\mathbf{n}(\theta) = (\tau_2, -\tau_1),$$

where the subscript θ of the position vector $\mathbf{X}$ denotes the derivative with respect to θ as $\mathbf{X}_\theta := \frac{d\mathbf{X}}{d\theta}$, and the stretching element is $\|\mathbf{X}_\theta\| = \sqrt{X_\theta^2 + Y_\theta^2}$. This notation will also be used for other variables throughout this paper. The signed curvature of the curve is obtained by

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

$$\kappa(\theta) = \frac{X_\theta Y_{\theta\theta} - Y_\theta X_{\theta\theta}}{\|\mathbf{X}_\theta\|^3} \tag{2.1}$$

and will be used later for the curvature-dependent reparametrization and for the application in mean curvature flow. We note that the above curvature can be computed in spectral accuracy by Fourier approximation of $\mathbf{X}$.

For the application of Willmore flow, we need to express the full expansion of the Laplace–Beltrami operator of curvature. Thereby, the term κ_{ss} and its related variables can also be written as

$$\kappa_{ss} := \frac{1}{\|\mathbf{X}_\theta\|}\frac{\partial}{\partial\theta}\left(\frac{1}{\|\mathbf{X}_\theta\|}\frac{\partial\kappa}{\partial\theta}\right) = \frac{-\mathbf{X}_\theta\cdot\mathbf{X}_{\theta\theta}}{\|\mathbf{X}_\theta\|^4}\frac{\partial\kappa}{\partial\theta} + \frac{1}{\|\mathbf{X}_\theta\|^2}\frac{\partial^2\kappa}{\partial\theta^2}, \tag{2.2}$$

where

$$\frac{\partial\kappa}{\partial\theta} = (X_\theta Y_{\theta\theta\theta} - Y_\theta X_{\theta\theta\theta})\|\mathbf{X}_\theta\|^{-3} - 3\kappa(\mathbf{X}_\theta\cdot\mathbf{X}_{\theta\theta})\|\mathbf{X}_\theta\|^{-2},$$

$$\begin{aligned}\frac{\partial^2\kappa}{\partial\theta^2} =& (X_{\theta\theta}Y_{\theta\theta\theta} + X_\theta Y_{\theta\theta\theta\theta} - Y_{\theta\theta}X_{\theta\theta\theta} - Y_\theta X_{\theta\theta\theta\theta})\|\mathbf{X}_\theta\|^{-3}\\ &- 3(X_\theta Y_{\theta\theta\theta} - Y_\theta X_{\theta\theta\theta})(\mathbf{X}_\theta\cdot\mathbf{X}_{\theta\theta})\|\mathbf{X}_\theta\|^{-5} - 3\kappa_\theta(\mathbf{X}_\theta\cdot\mathbf{X}_{\theta\theta})\|\mathbf{X}_\theta\|^{-2}\\ &- 3\kappa(\mathbf{X}_\theta\cdot\mathbf{X}_{\theta\theta\theta} + \|\mathbf{X}_{\theta\theta}\|^2)\|\mathbf{X}_\theta\|^{-2} + 6\kappa(\mathbf{X}_\theta\cdot\mathbf{X}_{\theta\theta})^2\|\mathbf{X}_\theta\|^{-4}.\end{aligned}$$

2.2. Equi-arclength reparametrization. For a closed curve $\mathbf{X}(\theta)$ over $\theta \in [0, 2\pi)$, the 2π-normalized arclength function of the curve is defined by

$$s(\theta) = 2\pi\int_0^\theta \|\mathbf{X}_\alpha\|\,\mathrm{d}\alpha \Big/ \int_0^{2\pi}\|\mathbf{X}_\alpha\|\,\mathrm{d}\alpha. \tag{2.3}$$

Here the 2π-normalization is only for convenience. Then constructing an equi-arclength parametrization with respect to $\mathbf{X}$ is up to finding the inverse function $s^{-1}(\theta)$. In principle, the inverse can be easily obtained by the exchange of variables between s and θ. As illustrated in Figure 1, the inverse function is the reflection of the normalized arclength function along the straight line, $s(\theta) = \theta$. However, the analytic form of inverse function is not available in practice. Therefore, in this paper, we

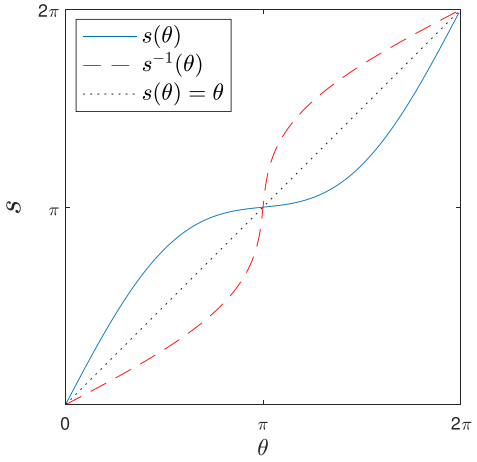

FIG. 1. *A 2π-normalized arclength function $s(\theta)$ and its inverse function $s^{-1}(\theta)$ are reflective on the straight line $s(\theta) = \theta$.*

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

attempt to find the numerical inverse of the 2π-normalized arclength function within spectral accuracy so that an equi-arclength reparametrization can be achieved. In differential geometry [20, Chap. 2], such a curve parameterized by arclength is usually called a unit-speed curve.

2.3. Equi-arclength-like reparametrization with monitor function. As a generalization of the equi-arclength parametrization described in the previous subsection, we can further consider the equi-arclength-like reparametrization with monitor function which is closely related to the adaptive mesh method [17]. The 2π-normalized arclength-like function is defined by

$$s(\theta) = 2\pi \int_0^\theta M(\alpha)\|\mathbf{X}_\alpha\|\,\mathrm{d}\alpha \Big/ \int_0^{2\pi} M(\alpha)\|\mathbf{X}_\alpha\|\,\mathrm{d}\alpha, \tag{2.4}$$

where the monitor (or weight) function $M(\alpha) > 0$ is chosen so that the arclength-like function increases monotonically. Similar to equi-arclength reparametrization, the inverse function s^{-1} of (2.4) will give the equi-arclength-like reparametrization. When $M = 1$, this function reduces to the arclength function written in (2.3). One can design an appropriate monitor function to suit a special purpose. For instance, a curvature-dependent function defined by

$$M(\alpha) = |\kappa|^{1/2} + C, \qquad \text{where} \quad C = \frac{\int_0^{2\pi} |\kappa|^{1/2}\|\mathbf{X}_\alpha\|\,\mathrm{d}\alpha}{\int_0^{2\pi} \|\mathbf{X}_\alpha\|\,\mathrm{d}\alpha}, \tag{2.5}$$

is used in [17] for forced curve shortening flow.

3. Numerical method. In this section, we describe the algorithm that obtains the equally redistributed points of a discretized curve by solving the equi-arclength (-like) reparametrization numerically described in the previous section. To achieve the spectral accuracy of the numerical solution, we use the Fourier approximation, the spectral differentiation, and the spectral integration for associated variables. We further demonstrate the availability of increment/decrement of point number in our method, which remeshes the discretized curve.

3.1. Equi-arclength redistribution of points. For given N points as an approximation of a closed curve, we set the constant angle width by $\Delta\theta = 2\pi/N$ and the corresponding equispaced angle by $\theta_j = (j-1)\Delta\theta$ for $j = 1, \ldots, N$. This is the initial parametrization of a given curve. Referring to (2.3), the 2π-normalized discrete arclength function is obtained from

$$s(\theta_j) = 2\pi \int_0^{\theta_j} \|\mathbf{X}_\alpha(\alpha)\|\,\mathrm{d}\alpha \Big/ \int_0^{2\pi} \|\mathbf{X}_\alpha(\alpha)\|\,\mathrm{d}\alpha, \tag{3.1}$$

where the stretching element is computed by $\|\mathbf{X}_\alpha(\alpha_j)\| = \sqrt{X_\alpha^2(\alpha_j) + Y_\alpha^2(\alpha_j)}$ after applying spectral differentiations to X and Y. For later use in this algorithm, we keep the Fourier coefficients $\widehat{X}$ and $\widehat{Y}$. To compute the arclength values for all j, spectral integration is used to obtain spectral accuracy. The details on spectral calculus are described in subsection 3.3.

We now need to find the discrete inverse function $s^{-1}(\theta_j)$ for all j which is equivalent to the new angle θ_j' satisfying the equi-arclength redistribution; see Figure 1. To do this, as similarly set for the initial angle, we set the constant arclength width by $\Delta s = 2\pi/N$ and the corresponding equispaced arclength by $s_j = (j-1)\Delta s$ for $j = 1, \ldots, N$. We remark that the given N points can be transformed to any desired

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

number of points for curve remeshing, which will be discussed in subsection 3.5. Then for fixed equispaced s_j, we seek the new angle θ_j' satisfying the equation

$$s_j = 2\pi \int_0^{\theta_j'} \|\mathbf{X}_\alpha(\alpha)\| \, \mathrm{d}\alpha \Big/ \int_0^{2\pi} \|\mathbf{X}_\alpha(\alpha)\| \, \mathrm{d}\alpha := s(\theta_j') \tag{3.2}$$

by root finding. We note that the function $s(\theta_j')$ can be modified to be a 2π-periodic function so that the Fourier approximation is used to achieve spectral accuracy; see the details in subsection 3.4. As the final step of our method, we directly compute the inverse DFT to update the position vector $\mathbf{X}$ by using the new angle θ_j' referring to (3.6) as

$$\mathbf{X}(\theta_j') = \sum_{k=-N/2}^{N/2-1} \widehat{\mathbf{X}}(k) e^{ik\theta_j'}, \tag{3.3}$$

where the coefficients $\widehat{\mathbf{X}} = (\widehat{X}, \widehat{Y})$ were already obtained from initially given points with equispaced angles in the beginning of this algorithm. It is clear that the updated position vector $\mathbf{X}(\theta_j')$ meets the equi-arclength redistribution of points, whenever θ_j' is chosen to hold (3.2) within machine precision. So, the overall accuracy of the present method also depends on the error of new angle in the root finding step.

3.2. Curvature-dependent redistribution of points. Referring to (2.4) and (2.5), in order to have curvature-dependent redistribution of points, we only need to replace the discrete arclength function (3.1) by

$$s(\theta_j) = 2\pi \int_0^{\theta_j} M(\alpha) \|\mathbf{X}_\alpha\| \, \mathrm{d}\alpha \Big/ \int_0^{2\pi} M(\alpha) \|\mathbf{X}_\alpha\| \, \mathrm{d}\alpha, \tag{3.4}$$

where the discretized monitor function associated with curvature-dependence is given by

$$M(\alpha_j) = |\kappa(\alpha_j)|^{1/2} + \int_0^{2\pi} |\kappa|^{1/2} \|\mathbf{X}_\alpha\| \, \mathrm{d}\alpha \Big/ \int_0^{2\pi} \|\mathbf{X}_\alpha\| \, \mathrm{d}\alpha.$$

As in the equi-arclength case, the spectral integration is used to compute the arclength-like values in (3.4). In computing curvature κ, we applied spectral differentiations to the position vector and then used (2.1). As done for the equi-arclength condition in (3.1), we seek new angle θ_j' over fixed equispaced s_j satisfying

$$s_j = 2\pi \int_0^{\theta_j'} M(\alpha) \|\mathbf{X}_\alpha\| \, \mathrm{d}\alpha \Big/ \int_0^{2\pi} M(\alpha) \|\mathbf{X}_\alpha\| \, \mathrm{d}\alpha := s(\theta_j'). \tag{3.5}$$

After performing the rest of the procedure illustrated in subsection 3.1, we obtain the curvature-dependent redistribution of points.

3.3. Spectral differentiation and integration. The Fourier approximation of a scalar function f or inverse DFT is

$$f(\theta_j) = \sum_{k=-N/2}^{N/2-1} \widehat{f}(k) e^{ik\theta_j}, \tag{3.6}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

where i is the imaginary number and $j = 1, \ldots, N$. Its Fourier coefficient or DFT is

$$\widehat{f}(k) = \frac{1}{N} \sum_{j=1}^{N} f(\theta_j) e^{-ik\theta_j},$$

where $k = -N/2, \ldots, N/2 - 1$. To the position vector $\mathbf{X} = (X, Y)$, we apply these transforms individually.

Based on Fourier representation, the nth differentiation of $f(\theta)$ is

$$\frac{\mathrm{d}^n f(\theta)}{\mathrm{d}\theta^n} = \sum_{k=-N/2}^{N/2-1} (ik)^n \widehat{f}(k) e^{ik\theta}. \tag{3.7}$$

After direct calculation [4], we also have the spectral integration as

$$\int_0^\theta f(\alpha)\,\mathrm{d}\alpha = \int_0^\theta \sum_{k=-N/2}^{N/2-1} \widehat{f}(k) e^{ik\alpha}\,\mathrm{d}\alpha = \widehat{f}(0)\theta + \sum_{k=-N/2}^{N/2-1} \widehat{g}(k) e^{ik\theta}, \tag{3.8}$$

where

$$\widehat{g}(k) = \begin{cases} -\displaystyle\sum_{j\neq 0} \frac{\widehat{f}(j)}{ij} & \text{if } k = 0, \\ \dfrac{\widehat{f}(k)}{ik} & \text{otherwise.} \end{cases}$$

For given points labeled with natural numbers, fast computation of spectral differentiation in (3.7) is done by applying the forward fast Fourier transform (FFT) to the function f to obtain the coefficient $\widehat{f}(k)$ and then the inverse FFT to the modified coefficient $(ik)^n \widehat{f}(k)$. The fast spectral integration of a function $f(\theta)$ in (3.8) is also accomplished by applying the inverse FFT to the coefficient $\widehat{g}(k)$, followed by adding $\widehat{f}(0)\theta$ to the result. To find total arclength(-like) value, we explicitly calculate the series formulation written in the rightmost part of (3.8) after replacing θ by 2π. For de-aliasing, the 2/3-rule filter is adopted.

3.4. Root finding. In finding a new angle (or discrete inverse function) as described in subsections 3.1 and 3.2, we use the false position method, a variant of the bisection method. This method guarantees finding the approximate solution when a proper interval is initially given. In our approach of root finding, the existence of an inverse function ensures that there exists a unique solution in $[0, 2\pi)$ [20].

For the equispaced normalized arclength(-like) value denoted by $s_j = (j-1)\Delta s$ with $\Delta s = 2\pi/N$, we seek new angle θ'_j which configures the equi-arclength(-like) redistribution of points satisfying

$$s_j = \theta'_j + \sum_{k=-N/2}^{N/2-1} \widehat{h}(k) e^{ik\theta'_j} := s(\theta'_j). \tag{3.9}$$

Here $\widehat{h}$ is the Fourier coefficients of a 2π-periodic function newly defined by $h(\theta_j) = s(\theta_j) - \theta_j$ over equispaced θ_j, where $s(\theta_j)$ given in (3.1) and (3.4) is computed by spectral integration. Then fixing s_j and $\widehat{h}$ in the above equation, the false position method solves $s_j = s(\theta'_j)$ by finding θ'_j within a specified precision. Note that as the

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

right-hand side of (3.2) and (3.5) is denoted by $s(\theta'_j)$, they are approximated by the series formulation in (3.9). In the implementation level of the false position method, we choose 100 as the maximum number of iterations and 10^{-15} as the error tolerance to stop the iteration.

3.5. Control of the number of points. In our method, it is an easy task to control the number of points, i.e., an increment and a decrement of the number of points under the equi-arclength(-like) condition. Let us choose an integer M, the desired number of points. Our code is rarely modified to obtain the M-number of points from the given points. To do this, in finding roots as described in subsection 3.4, we find the new angle θ'_j for $s_j = (j-1)\Delta s$, where $\Delta s = 2\pi/M$ and $j = 1, \ldots, M$. After directly computing the inverse DFT in (3.3) with θ'_j, we immediately obtain the remeshed curve having M number of points.

In subsection 4.4, we provide some numerical results of equi-arclength remeshing for points perturbed smoothly on the unit circle. Then in subsection 5.2, a star-shaped curve in mean curvature flow is studied under both types of remeshing, where the decrement in number of points mainly occurs. To see the effect of point-number increment, both types of remeshing are applied to shear-induced droplet deformation in subsection 5.4. To determine the desired number of points in terms of time, we use

$$M(t) = \left\lceil \frac{L(t)}{L(0)/M(0)} \right\rceil, \tag{3.10}$$

where $L(t)$ is the total arclength of a given curve at time t and $M(0)$ is the initial number of points. By doing so, the deforming curve in time keeps the almost constant arclength in the case of equi-arclength remeshing; see Figures 14(a) and 17(a). In the case of curvature-dependent remeshing, the above relation (3.10) is also used. This strategy enables AMR when required; see Figure 17(b).

3.6. Summary of the algorithm and its cost. We end this section by providing a brief step-by-step description of our numerical algorithm and its computational cost.

Step 1. Compute the Fourier coefficients of a given curve for each spatial component.

Step 2. Using spectral differentiations, compute the derivatives of the position vector $\mathbf{X}$ and then obtain the stretching element $\|\mathbf{X}_\theta\|$. In the case of curvature-dependent redistribution, compute the related monitor function as well.

Step 3. Obtain the 2π-normalized discrete arclength(-like) function by using spectral integration.

Step 4. By root finding over equispaced arclength(-like) value, obtain the new angle θ' which is exactly the discrete inverse function with respect to equispaced angle θ; see Figure 1.

Step 5. By inserting the new angle θ' one by one into the Fourier representation (3.3) with the Fourier coefficients obtained in Step 1, compute the position vector $\mathbf{X}(\theta')$ which gives the equi-arclength(-like) redistribution of points.

The computational cost in Steps 1–3 for N-number of points is mainly dependent on the FFT operations, thus $O(N \log N)$. In Step 4, for each iteration, it costs $O(N^2)$ operations to directly compute the Fourier series in (3.9), so this step has the highest computational cost in our algorithm. Again in Step 5, the direct computing of Fourier series at N angles requires $O(N^2)$ operations, but done only once with the solution of new angles. Thus, the overall cost of our algorithm is $O(N^2)$ due to Steps 4 and 5 and is strongly dependent on the number of iterations for finding new angles in

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Step 4. We will show later that the average number of iterations is almost constant for different numbers of points.

4. Numerical accuracy and convergence rate. In this section, we investigate the accuracy and the convergence rate of our numerical algorithm for points distributed on the unit circle. The applications of our method for complex curves and dynamic flows will be shown in the next section. The circle is chosen in this section because the exact solution under an equi-arclength condition is obvious, so the error can be easily computed. We test three cases where points are initially perturbed by smooth angles, nonsmooth angles, and randomly chosen angles. In addition, the remeshing of a curve by controlling the number of points is performed.

Before the tests, we mention the norms and the convergence rates used in this paper. For a vector field $\mathbf{V}(\theta) = (U(\theta), V(\theta))$, the L_∞ norm is defined by

$$\|\mathbf{V}\|_\infty = \max_i \left\{ \left(U^2(\theta_i) + V^2(\theta_i) \right)^{1/2} \right\}$$

and the L_2 norm on the unit circle by

$$\|\mathbf{V}\|_2 = \left(\frac{1}{N} \sum_{i=1}^{N} \left(U^2(\theta_i) + V^2(\theta_i) \right) \right)^{1/2},$$

where N is the number of points. In general except the unit circle, we use

$$\|\mathbf{V}\|_2 = \left(\frac{1}{2\pi} \sum_{i=1}^{N} \left(U^2(\theta_i) + V^2(\theta_i) \right) \Delta s_i \right)^{1/2},$$

where the computed arclength element is $\Delta s_i = \|\mathbf{X}_\theta(\theta_i)\| \Delta\theta$ with $\Delta\theta = 2\pi/N$ for $i = 1, \ldots, N$. So the errors of the position vector $\mathbf{X}$ with its length N in L_∞ and L_2 are respectively defined by

$$E_\infty^N = \|\mathbf{X}_N - \mathbf{X}_e\|_\infty, \qquad E_2^N = \|\mathbf{X}_N - \mathbf{X}_e\|_2,$$

where $\mathbf{X}_e$ is the exact solution or the numerical solution obtained in a finer resolution. The convergence rate of the position vector $\mathbf{X}$ in L_∞ norm is obtained from

$$\text{Rate} = \log_2 \left(E_\infty^N / E_\infty^{2N} \right),$$

and the rate in L_2 norm is similarly obtained. When a numerical solution obtained in a finer resolution is used instead of the exact solution, the error is computed after the projection of numerical solution onto the coarse resolution.

4.1. Smooth perturbation. For N-number of points fixed by $\mathbf{X}(0) = (1, 0)$, the unit circle written by

$$\mathbf{X}(\theta) := (X(\theta), Y(\theta)) = (\cos\theta, \sin\theta)$$

has the equi-arclength distribution of points only when the angle (or parameter) θ is equispaced by $\theta_i = (i-1)2\pi/N$ for $i = 1, \ldots, N$.

In this test, we initially perturb the points by replacing the angle θ by

$$\tilde{\theta} = \theta + 0.25 \sin(\omega\theta), \tag{4.1}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

so the points perturbed smoothly on the unit circle are obtained from $\mathbf{X}(\tilde{\theta}) = (\cos\tilde{\theta}, \sin\tilde{\theta})$. Onto these points, we apply our algorithm to obtain the equi-arclength redistribution described in section 3, specifically in subsection 3.1. We tested for different frequencies $\omega = 1, 2, 3$ in (4.1) and different numbers of points $N = 16, 32, 64, 128$. In Table 1, the spectral accuracy is clearly observed for $\omega = 1, 2, 3$ respectively when $N = 32, 64, 128$. This identifies that the more perturbed, the more points required to achieve spectral accuracy, which is an inherent property of Fourier approximation. A qualitative result when $\omega = 3$ and $N = 32$ is shown in Figure 2, where the exact solution is compared with the numerical solution obtained from the initially perturbed points. One can clearly see that the numerical solution using our method agrees well with the exact solution.

As described in subsection 3.4, the computational cost of our algorithm is highly dependent on the number of iterations in the root finding step, since the inverse DFT is directly computed. In Figure 3, the average number of iterations is plotted with respect to number of points N, where different frequencies ω are used. This corresponds to the cases shown in Table 1. We observe that as the frequency ω or the complexity of the curve increases, the average number of iterations increases as predictable. Fortunately, however, at fixed ω the increment of average number of iterations is relatively small compared with that of number of points. This implies that the average number of iterations is almost independent of number of points, once the complexity of the curve is fixed. So, we can conclude that the actual computational cost of our method is still $O(N^2)$ for a given stationary curve as discussed in subsection 3.6.

TABLE 1

Convergence analysis of the position vector $\mathbf{X}$ *in* L_∞ *and* L_2 *norms in the case of smooth perturbation with different frequencies* $\omega = 1, 2, 3$ *in* (4.1). *An instance with* $N = 32$ *is shown in Figure* 2.

N	E_∞^N	E_2^N	E_∞^N	E_2^N	E_∞^N	E_2^N
	($\omega = 1$)		($\omega = 2$)		($\omega = 3$)	
16	1.040E-05	7.523E-06	7.792E-03	6.411E-03	1.555E-02	1.150E-02
32	2.076E-14	1.474E-14	2.651E-07	2.438E-07	3.445E-04	2.407E-04
64	1.299E-15	6.028E-16	1.053E-15	5.175E-16	1.044E-10	9.185E-11
128	2.512E-15	6.289E-16	1.790E-15	6.440E-16	3.975E-15	9.121E-16

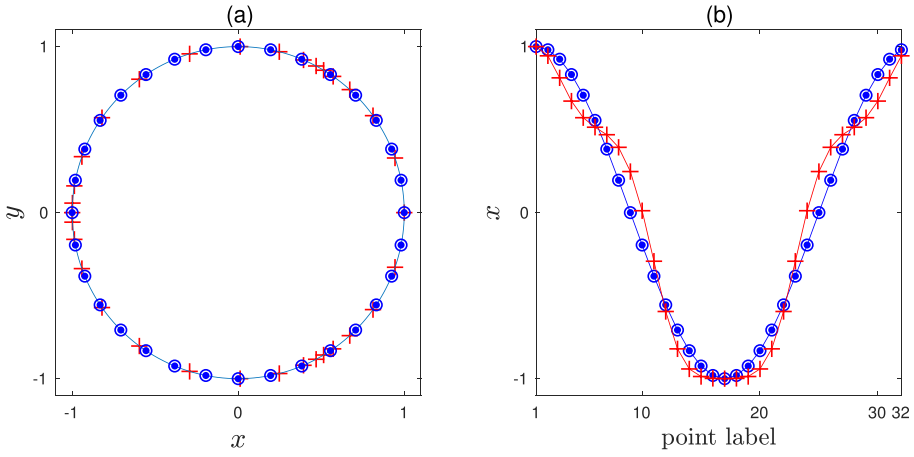

FIG. 2. *In the case of smooth perturbation with the frequency* $\omega = 3$*:* (a) *given* 32 *points (red +), exact solution (blue ◯), and numerical result (blue •),* (b) *the x-position function of points with respect to point label.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

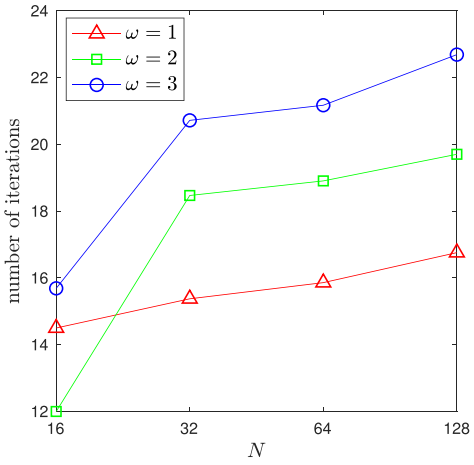

FIG. 3. *In the case of smooth perturbation, the average number of iterations in root finding.*

TABLE 2

Convergence analysis of the position vector $\mathbf{X}$ *in* L_∞ *and* L_2 *norms in the case of nonsmooth perturbation written in* (4.2). *An instance with* $N = 32$ *is shown in Figure* 4.

N	E_∞^N	Rate	E_2^N	Rate
64	5.645E-02	-	4.051E-02	-
128	3.259E-02	0.79	2.246E-02	0.85
256	1.875E-02	0.79	1.244E-02	0.85
512	1.068E-02	0.81	6.905E-03	0.84
1024	5.915E-03	0.85	3.811E-03	0.85

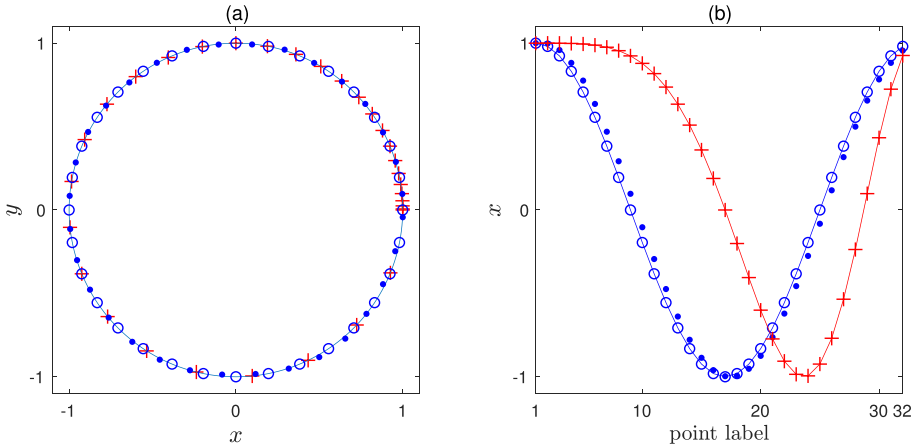

FIG. 4. *In the case of nonsmooth perturbation at* $\theta = 0$*:* (a) *given* 32 *points (red +), exact solution (blue ◯), and computed result (blue •),* (b) *the x-position function of points with respect to point label.*

4.2. Nonsmooth perturbation. Unlike the preceding test, we initially perturb the points by replacing the equispaced angle θ by

$$\tilde{\theta} = \theta^2/(2\pi), \tag{4.2}$$

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

which gives nonsmooth distribution of points at $\theta = 0$ or 2π. As shown in Table 2, the error is far from machine precision. This is predictable since the Fourier representation of the nonsmooth function has large error near the nonsmooth region, known as the Gibbs phenomenon. A case with $N = 32$ is plotted in Figure 4, where the exact solution is compared with the numerical solution obtained from the initially perturbed points. It shows that the numerical solution using our method does not agree well with the exact solution. Fortunately, as seen in the table, it still converges with a roughly first-order convergence rate, regardless of norms. This indicates that our method is definitely applicable to practical problems. To support this conclusion, curve motion in various flows will be studied in section 5.

4.3. Random perturbation. To test an extreme case, we choose the angles randomly over $[0, 2\pi)$ so that the points are distributed randomly on the unit circle. For computational convenience beforehand, a point at $\theta = 0$ is always included and the chosen points are ordered by angle. As already indicated in the case of nonsmooth perturbation, we never observe spectral accuracy in this test as shown in Table 3. The written error is the mean value obtained from 100 samples for each case, and its standard deviation is provided in parentheses. Surprisingly, we observe in the table that our numerical solutions of position vector converge with the half-order. The corresponding standard deviations also decrease as the number of points increases. This confirms that our method works successfully even for randomly given points. In Figure 5, an example with $N = 32$ is shown. One can see that our numerical result gives quite well-distributed points under the equi-arclength condition, although the initial distribution of points is severely nonsmooth overall.

TABLE 3

Statistical convergence analysis of the position vector $\mathbf{X}$ *in* L_∞ *and* L_2 *norms in the case of random perturbation with* 100 *samples. The mean value and its corresponding standard deviation in parentheses are given. An instance with* $N = 32$ *is shown in Figure* 5.

N	E_∞^N		Rate	E_2^N		Rate
64	9.828E-02	(±4.506E-02)	-	5.291E-02	(±2.954E-02)	-
128	7.235E-02	(±3.218E-02)	0.44	3.570E-02	(±1.806E-02)	0.56
256	4.914E-02	(±1.808E-02)	0.55	2.255E-02	(±1.017E-02)	0.66
512	3.394E-02	(±1.381E-02)	0.53	1.584E-02	(±8.247E-03)	0.50
1024	2.615E-02	(±9.133E-03)	0.37	1.177E-02	(±5.179E-03)	0.42

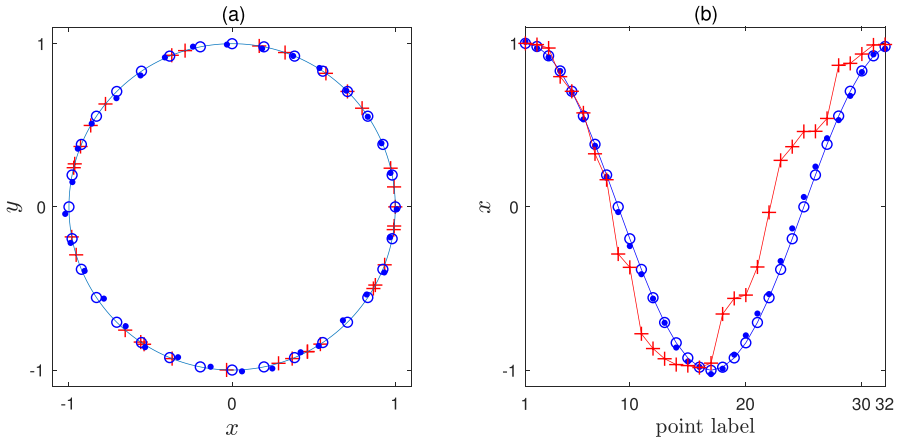

FIG. 5. *An instance of random perturbation:* (a) *given* 32 *points (red +), exact solution (blue ◯), and computed result (blue •),* (b) *the x-position function of points with respect to point label.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

4.4. Decrement/increment of the number of points. So far, we have fixed the number of points during application of the equi-arclength redistribution. We now test the availability of our method in controlling the number of points, the so-called remeshing technique. To do so, we choose the initial data by the points used in the case of smooth perturbation with fixed $\omega = 3$ and $N = 16, 32, 64, 128$ shown in subsection 4.1. We decrease and increase the number of points N to the desired number of points $M = 13$ and 131, respectively. As described in subsection 3.5, we shall find only M-number of new angles, rather than N, with respect to equispaced arclength function. Then the rest of the procedure is the same as for N-number of new angles. In Table 4, interestingly regardless of N, our numerical results for both $M = 13$ and 131 are quite comparable with the ones shown in Table 1, meaning that the remeshed curve has no loss in accuracy. So it appears that the accuracy of our approach is entirely independent of the desired number of points. In Figure 6, the numerical results of remeshed curve with $N = 32$ are shown. For both cases of the decrement ($M = 13$) and the increment ($M = 131$), the exact solutions perfectly agree with our numerical solutions obtained from the smoothly perturbed points.

5. Numerical results. So far, we have fixed the curve to be the unit circle since the exact solution is easily obtained. In this section, to demonstrate the versatility of our method, various curves are examined by comparison with numerical solution obtained in a finer resolution. It then follows by a series of numerical tests on curve motion in mean curvature flow, Willmore flow, and Stokes flow.

For all tests carried out in this section, we apply the equi-arclength and curvature-dependent redistributions described in subsections 3.1 and 3.2, respectively. The initially given curves are approximated by N-number of points over equispaced angles by $\theta_i = (i-1)2\pi/N$ for $i = 1, \ldots, N$. We provide both the quantitative and qualitative results for fixed closed curves and a fixed periodic planar curve, but only the qualitative result for evolving closed curves in dynamic flow.

5.1. Fixed curves. We apply our method to fixed closed curves such as an ellipse, a star-shaped curve, and a more complex curve. A periodic planar curve is tested as well.

5.1.1. An ellipse with aspect ratio 6 : 1. Apart from a circle, an ellipse is a highly compelling candidate in our study of a fixed curve, because an ellipse is often used in theory as an approximated solution and in practice as initial data. We use an ellipse with aspect ratio 6 : 1 defined by

$$(X(\theta), Y(\theta)) = (6\cos\theta, \sin\theta), \tag{5.1}$$

where θ is initially equispaced and new values shall be found to satisfy equi-arclength or curvature-dependent redistribution of points.

In Table 5, the almost machine precision is achieved when 512 points are used. Compared to the unit circle observed in Table 1, the ellipse with aspect ratio 6 : 1 requires more points to obtain spectral accuracy. As illustrated in Figure 7, one can see that our numerical results agree qualitatively well with the numerical solution obtained in a finer resolution with $N = 2048$. We observe in the table that for fixed N, the accuracy in the case of curvature-dependent redistribution is lower than in the case of equi-arclength redistribution. We attribute this to dependence of sufficient points (or frequency modes) to Fourier approximation of discrete arclength(-like) function used in (3.9). This idea will be strengthened below in successive tests.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

TABLE 4

Decrement/increment of number of points from N to the desired number of points $M = 13$ and 131. An instance with $N = 32$ is shown in Figure 6.

N	E_∞^N	E_2^N	E_∞^N	E_2^N
	($M = 13$)		($M = 131$)	
16	1.555E-02	1.082E-02	1.558E-02	1.124E-02
32	3.445E-04	2.329E-04	3.445E-04	2.392E-04
64	1.032E-10	9.159E-11	1.060E-10	9.195E-11
128	2.065E-15	9.938E-16	2.742E-15	8.625E-16

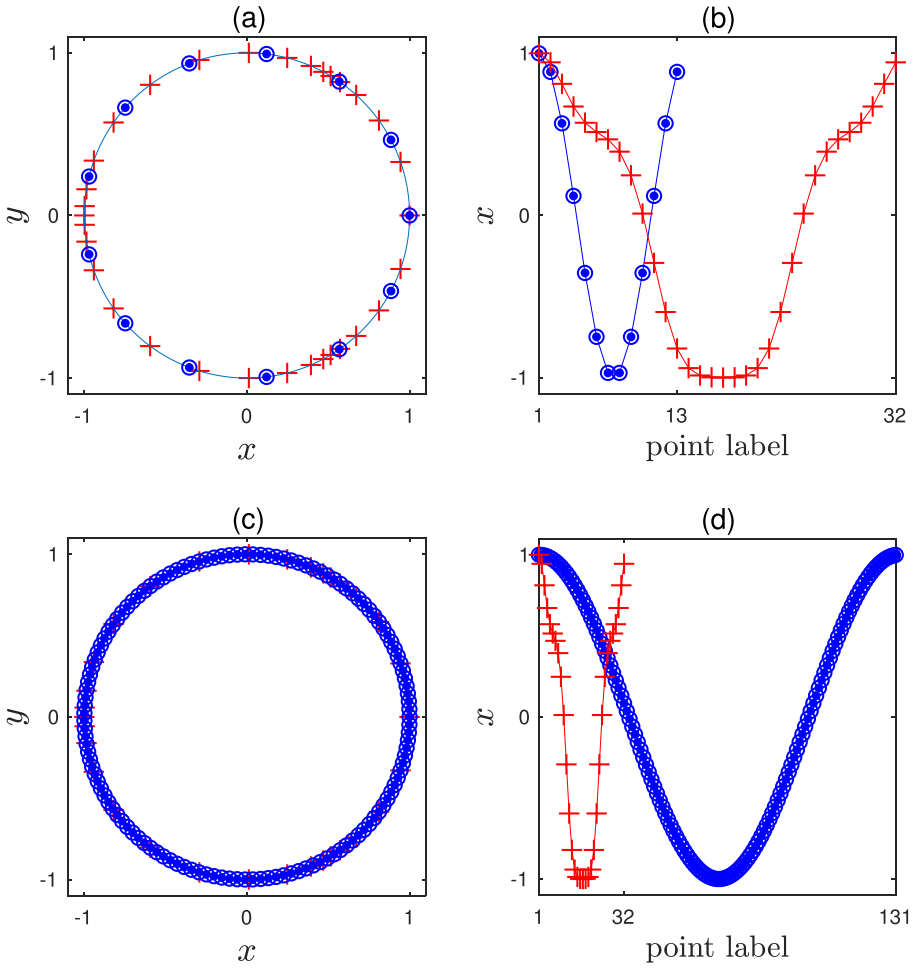

FIG. 6. *Control of the number of points:* (a), (b) *decrement of number of points from* 32 *to* 13, (c), (d) *increment of number of points from* 32 *to* 131, *given points (red +), exact solution (blue ◯), and computed result (blue •). This corresponds to the case of smooth perturbation with the frequency* $\omega = 3$.

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

TABLE 5

For an ellipse defined in (5.1)*, convergence analysis of the position vector* $\mathbf{X}$ *in* L_∞ *and* L_2 *norms. Here the equi-arclength redistribution (Equi.) and the curvature-dependent redistribution (Curv.) are tested. The numerical solution* $\mathbf{X}_e$ *is obtained by using* 2048 *points. The cases with* $N = 64$ *correspond to Figure* 7.

N	E_∞^N	E_2^N	E_∞^N	E_2^N
	(Equi.)		(Curv.)	
64	5.345E-05	5.649E-05	1.851E-04	2.594E-04
128	6.578E-07	3.992E-07	2.289E-06	2.977E-06
256	7.510E-11	4.491E-11	4.310E-10	5.309E-10
512	1.251E-14	6.003E-15	1.200E-13	7.587E-14

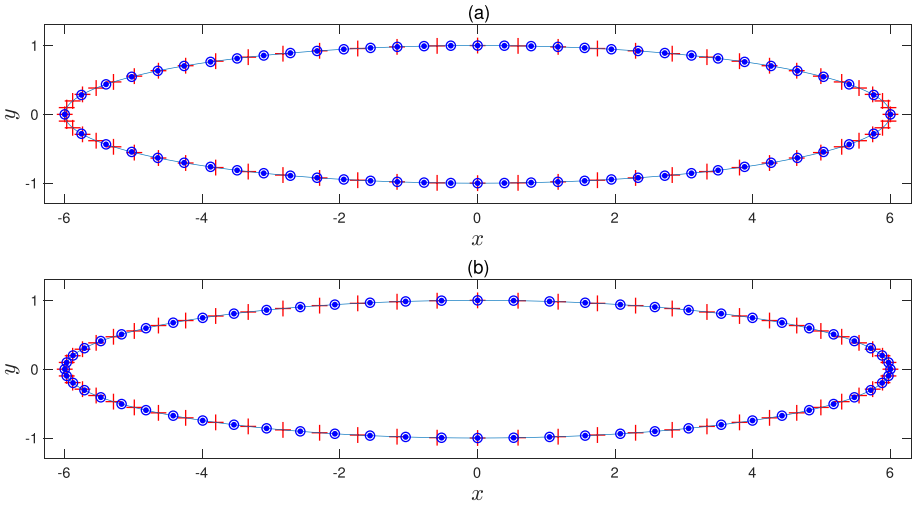

FIG. 7. *An ellipse defined in* (5.1)*:* (a) *equi-arclength redistribution,* (b) *curvature-dependent redistribution, given* 64 *points (red +), numerical solution obtained from* 2048 *points (blue ◯), and computed result from the given points (blue •).*

5.1.2. A star-shaped curve. We now test for a star-shaped curve given by

$$(x, y) = (1 + 0.5\sin(5\theta))\,(\cos\theta, \sin\theta). \tag{5.2}$$

This curve has higher curvature compared to the ellipse tested above.

In Table 6, the errors are shown for $N = 64, 128, 256, 512, 1024, 2048$. Machine precision is obtained for $N = 2048$ in the case of equi-arclength redistribution, while it is never obtained in the case of curvature-dependent redistribution. Compared to the results obtained for the ellipse above, more points are required for the star-shaped curve to obtain a fixed desired accuracy, no matter what redistribution is used. From these observations, one can expect that the number of points needed to achieve a specified accuracy depends not only on the type of redistribution but also on the complexity of the initially given curve. This is not surprising as seen in subsection 4.1, since both the curve position vector and its arclength function are approximated by truncated Fourier series. We will observe similar behavior in the next test for a complex curve. Before ending, we plot an example with $N = 64$ in Figure 8, in which the numerical result agrees well with the solution under the equi-arclength condition, while it disagrees somewhat under the curvature-dependent condition. This illustrates the tendency mentioned here and in the case of the ellipse.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

TABLE 6

For a star-shaped curve defined in (5.2)*, convergence analysis of the position vector* $\mathbf{X}$ *in* L_∞ *and* L_2 *norms. Here the equi-arclength redistribution (Equi.) and the curvature-dependent redistribution (Curv.) are tested. The numerical solution* $\mathbf{X}_e$ *is obtained by using* 8192 *points. The cases with* $N = 64$ *correspond to Figure* 8.

N	E_∞^N	E_2^N	E_∞^N	E_2^N
	(Equi.)		(Curv.)	
64	4.633E-03	2.608E-03	6.089E-02	4.007E-02
128	4.500E-04	2.359E-04	8.349E-03	4.888E-03
256	2.466E-05	8.770E-06	4.915E-03	2.952E-03
512	1.505E-07	5.072E-08	1.816E-03	8.596E-04
1024	3.711E-11	1.004E-11	1.077E-04	5.794E-04
2048	9.390E-15	3.487E-15	1.955E-04	8.345E-05

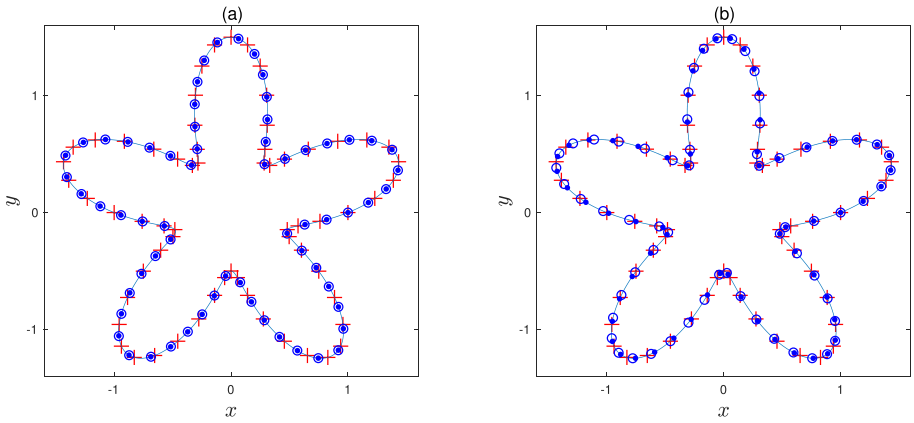

FIG. 8. *A star-shaped curve defined in* (5.2)*:* (a) *equi-arclength redistribution,* (b) *curvature-dependent redistribution, given* 64 *points (red +), numerical solution obtained from* 8192 *points (blue ◯), and computed result from the given points (blue •).*

5.1.3. A complex curve. We test a more complex curve defined by

$$(x, y) = \left(\cos\theta,\ 0.5\sin\theta + \sin(\cos\theta) + (0.2 + \sin\theta\sin^2(3\theta))\sin\theta\right). \tag{5.3}$$

This curve is also used in other papers [19, 17].

In a comparison between results in Tables 6 and 7, the overall errors are almost comparable to each other. So, this result supports the argument presented in the case of a star-shaped curve. As a result, it suggests to choose a proper number of points by taking the desired accuracy, the complexity of the curve itself, and the type of redistribution into account as a whole. In Figure 9, an example with $N = 64$ is plotted. As discussed above, our numerical result agrees closely with the solution in the case of equi-arclength redistribution, while it disagrees at many points in the case of curvature-dependent redistribution. This again indicates that, in practice, a considerable number of points may be used to achieve spectral accuracy under the curvature-dependent condition.

5.1.4. Periodic planar curve. As the last test of this subsection, we apply our method to a 2π-periodic planar curve defined by

$$(X(\theta), Y(\theta)) = (\theta + 2\sin\theta,\ 0.5\sin\theta), \tag{5.4}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

TABLE 7

For a complex curve defined in (5.3)*, convergence analysis of the position vector* $\mathbf{X}$ *in* L_∞ *and* L_2 *norms. Here the equi-arclength redistribution (Equi.) and the curvature-dependent redistribution (Curv.) are tested. The numerical solution* $\mathbf{X}_e$ *is obtained by using* 8192 *points. The cases with* $N = 64$ *correspond to Figure* 9.

N	E_∞^N	E_2^N	E_∞^N	E_2^N
	(Equi.)		(Curv.)	
64	6.083E-03	2.851E-03	2.730E-02	1.538E-02
128	4.566E-04	1.883E-04	2.175E-02	1.069E-02
256	9.849E-06	2.745E-06	6.725E-03	3.757E-03
512	3.575E-08	7.106E-09	1.337E-03	6.918E-04
1024	1.101E-12	2.146E-13	4.863E-04	1.959E-04
2048	5.978E-15	1.690E-15	2.376E-04	9.288E-05

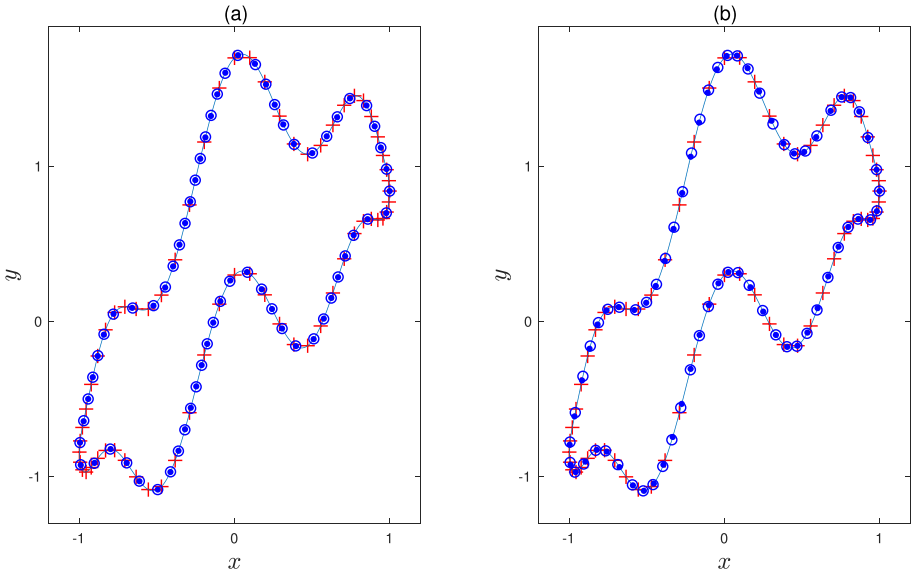

FIG. 9. *A complex curve defined in* (5.3)*:* (a) *equi-arclength redistribution,* (b) *curvature-dependent redistribution, given* 64 *points (red +), numerical solution obtained from* 8192 *points (blue ◯), and computed result from the given points (blue •).*

where $\theta \in [0, 2\pi)$ is an initial equi-spaced parameter, and we shall find a new angle satisfying equi-arclength or curvature-dependent redistribution of points. The numerical method presented in section 3 still works without any modification. In computing the first derivative of $X(\theta)$, however, care must be taken to achieve spectral accuracy. Suppose that the planar curve denoted by $(X(\theta), Y(\theta))$ has β-periodicity in the X-direction. From the relation $X = \widetilde{X} + \frac{\beta}{2\pi}\theta$ where $\widetilde{X}$ is a β-periodic function, one can use FFT for $\widetilde{X}$ instead of X. By applying spectral differentiations to $\widetilde{X}$, we obtain the derivatives of X by $\frac{dX}{d\theta} = \frac{d\widetilde{X}}{d\theta} + \frac{\beta}{2\pi}$ and $\frac{d^n X}{d\theta^n} = \frac{d^n \widetilde{X}}{d\theta^n}$ for $n \geq 2$. Then after applying our algorithm to $(\widetilde{X}(\theta), Y(\theta))$, we have $(\widetilde{X}(\theta'), Y(\theta'))$ with newly obtained angle θ', so the solution of X can be recovered from $X(\theta') = \widetilde{X}(\theta') + \frac{\beta}{2\pi}\theta'$.

In Table 8, machine precision is observed when the number of points used is larger than 512 under an equi-arclength condition. Compared to this, under a curvature-dependent condition, it is hard to achieve spectral accuracy although the error decreases with increasing N. This again confirms that the equi-arclength will work better in practice than the curvature-dependent one when the moderate number of

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

TABLE 8

For a 2π-periodic planar curve defined in (5.4)*, convergence analysis of the position vector* $\mathbf{X}$ *in* L_∞ *and* L_2 *norms. Here the equi-arclength redistribution (Equi.) and the curvature-dependent redistribution (Curv.) are tested. The numerical solution* $\mathbf{X}_e$ *is obtained by using* 8192 *points. The cases with* $N = 64$ *correspond to Figure* 10.

N	E_∞^N	E_2^N	E_∞^N	E_2^N
	(Equi.)		(Curv.)	
64	6.334E-05	2.332E-05	8.146E-03	5.143E-03
128	8.169E-07	2.444E-07	3.205E-03	1.837E-03
256	5.589E-10	1.586E-10	1.144E-03	6.529E-04
512	4.584E-15	1.869E-15	4.112E-04	2.298E-04
1024	5.416E-15	1.646E-15	1.537E-04	7.917E-05
2048	5.412E-15	1.795E-15	5.261E-05	2.568E-05

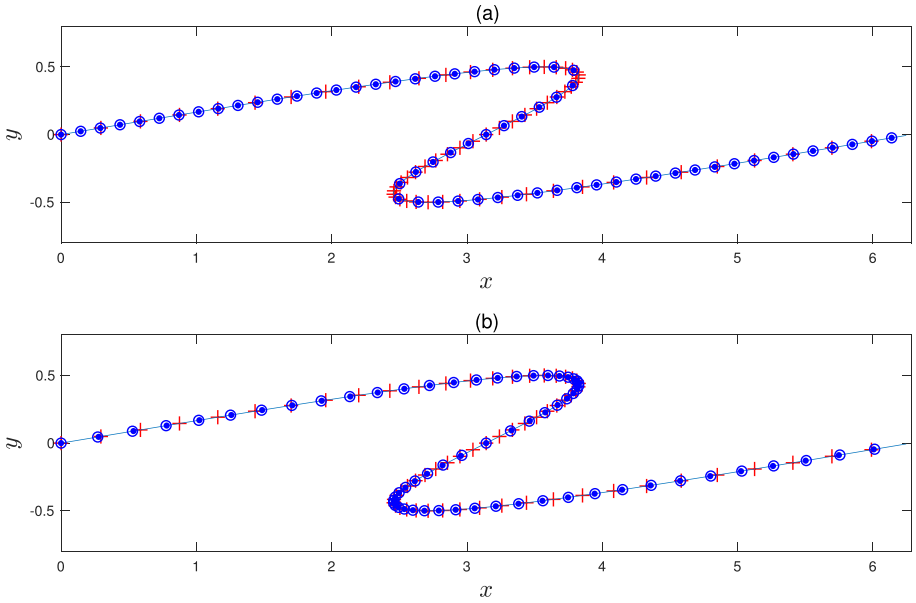

FIG. 10. *A 2π-periodic planar curve defined in* (5.4)*:* (a) *equi-arclength redistribution,* (b) *curvature-dependent redistribution, given* 64 *points (red +), numerical solution obtained from* 8192 *points (blue ◯), and computed result from the given points (blue •).*

points are used to approximate given curve. As shown in Figure 10, one can clearly observe that our numerical results even for the periodic planar curve agree qualitatively well with the numerical solution obtained in a finer resolution with $N = 8192$. So, the applications to elastic membrane [18] and hydroelastic wave propagation [5] will be promising candidates for a future research topic.

5.2. Mean curvature flow. From now on, we consider curves immersed in various flows to investigate the practical feasibility of our method in more general problems. As the first test, in this subsection, we study mean curvature flow of two curves, a star-shaped and a more complex, initially given by (5.2) and (5.3), respectively. The mean curvature flow (or curve shortening flow) is governed by

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

$$\frac{\partial \mathbf{X}}{\partial t} = -\kappa \mathbf{n},$$

where the signed curvature κ and the outward unit normal vector $\mathbf{n}$ are described in subsection 2.1. We discretize this equation by

$$\frac{\mathbf{X}_i^{n+1} - \mathbf{X}_i^n}{\Delta t} = -\kappa_i^n \mathbf{n}_i^n, \tag{5.5}$$

where the superscript n denotes the time step and the subscript i denotes the ith point on the curve. As seen, we simply use the forward Euler method for time integration, where the time step size is set by $\Delta t = 10^{-5}$. Approximating the curves with 128 points, the numerical results are obtained up to $t = 0.5$ for a star-shaped curve and $t = 0.34$ for a complex curve, respectively. The redistributions are applied initially once and every 100 time steps in both cases.

For comparison, we test the evolution of two curves in short time without applying the redistribution. As shown in Figure 11, for both curves, numerical instability occurs mainly due to the point clustering near the high curvature region. We now show that our method of point redistribution can contribute to avoid such instability. In Figures 12 and 13, we plot the time evolution of two curves with applying the redistribution. For both curves, its evolving motion equipped with the equi-arclength redistribution is shown in the left figure and that with the curvature-dependent one is in the right. We see that all curves in mean curvature flow converge to a circular curve following to the theoretical behavior of decrement in total arclength as time goes on. Refer to [19, Figure 1] and [17, Figure 8(b)] to compare with Figure 13(a) and also compare [17, Figure 8(c)] with Figure 13(b). One can see that the results obtained between two redistributions are qualitatively similar for all time steps, while in early times the deforming curves using the curvature-dependent redistribution give better approximation near high curvature regions.

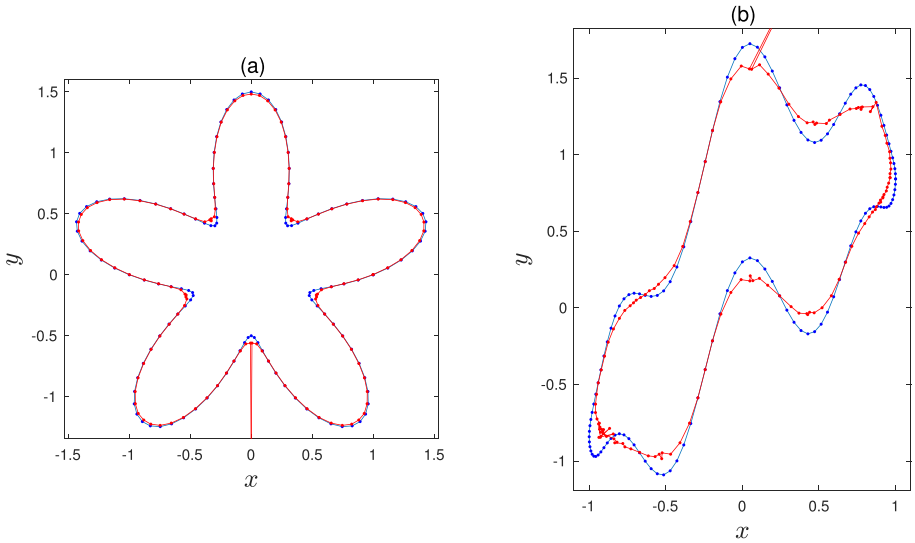

FIG. 11. *Without applying the redistribution, the mean curvature flow of two curves approximated with* 128 *number of points:* (a) *star-shaped curve defined in* (5.2), (b) *complex curve defined in* (5.3).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

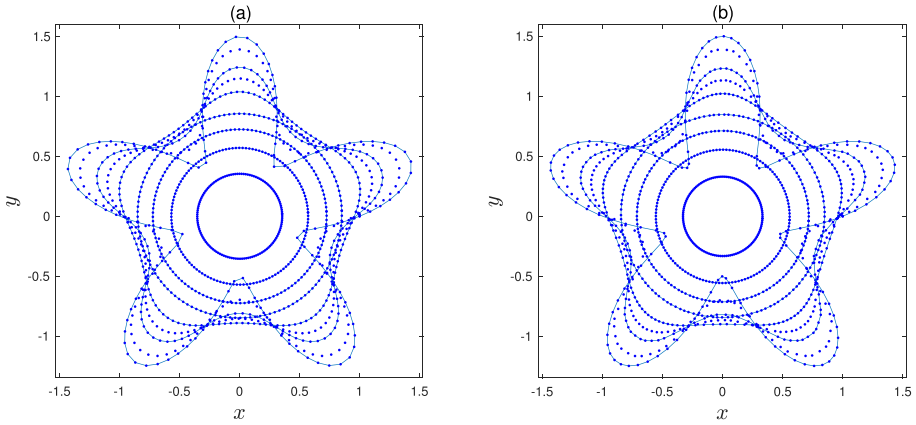

FIG. 12. *For some chosen times, the mean curvature flow of an initially star-shaped curve defined in* (5.2) *approximated with* 128 *points:* (a) *equi-arclength redistribution,* (b) *curvature-dependent redistribution. The redistribution is applied initially and every* 100 *time steps.*

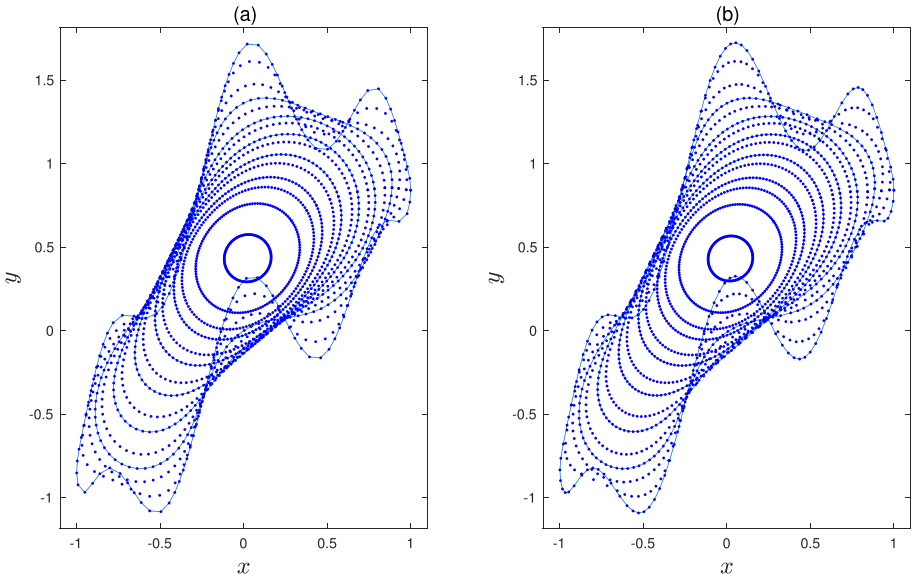

FIG. 13. *For some chosen times, snapshots of the mean curvature flow of an initially complex curve defined in* (5.3) *approximated with* 128 *points:* (a) *equi-arclength redistribution,* (b) *curvature-dependent redistribution. The redistribution is applied initially and every* 100 *time steps.*

Although not shown here, we found that the application of curvature-dependent redistribution in every time step may cause an oscillatory shape of the curve during time-dependent simulations. We attribute such behavior to the low accuracy of arclength(-like) function as seen in Table 6, because small error near high curvature regions would generate and accumulate large error later when curvature-dependent redistribution is often used. Thus, a strategy of applying at the proper time steps is suggested for the curvature-dependent one. Unlike this, the equi-arclength redistribution generally leads to a smooth solution regardless of its application period. For fixed time step size Δt, we also found that the simulations are more stable by using

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

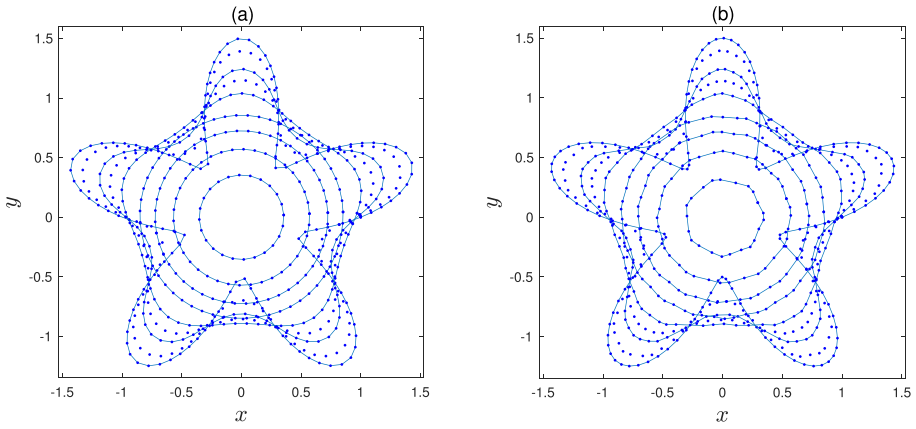

FIG. 14. *By applying remeshing, for some chosen times, the mean curvature flow of an initially star-shaped curve defined in* (5.2) *approximated with* 128 *points:* (a) *equi-arclength remeshing,* (b) *curvature-dependent remeshing. The remeshing is applied initially and every* 100 *time steps. This corresponds to Figure* 12.

the equi-arclength redistribution than by using the curvature-dependent one; refer [11] for related discussions.

To the star-shaped curve defined in (5.2), we now apply remeshing algorithm presented in subsection 3.5. Figure 14 shows its motion in mean curvature flow under both types of remeshing. In the case of the equi-arclength one (a), a smooth solution is obtained which is comparable to Figure 12(a), even though only 24 points remain to approximate the curve motion in the final time. Contrary to this, in the case of the curvature-dependent one (b), the curve has an oscillatory shape as the number of points decreases. A similar issue was raised when the curvature-dependent redistribution is often used even with fixed 128 points as discussed above. In the current test, however, we attribute such numerical instability to the inaccuracy of geometric quantities due to few frequency modes. We will show in subsection 5.4 that the curvature-dependent remeshing works well for an elongated droplet due to many points i.e., enough frequency modes.

5.3. Willmore flow. A curve in Willmore flow deforms to decrease the energy of total squared mean curvature, and the one-dimensional curve in Willmore flow is governed by

$$\frac{\partial \mathbf{X}}{\partial t} = \left(\kappa_{ss} + \frac{\kappa^3}{2}\right)\mathbf{n}, \tag{5.6}$$

where κ_{ss} is the Laplace–Beltrami operator of curvature as defined in (2.2). Using the forward Euler method, we discretize this equation by

$$\frac{\mathbf{X}_i^{n+1} - \mathbf{X}_i^n}{\Delta t} = \left((\kappa_{ss})_i^n + \frac{(\kappa_i^n)^3}{2}\right)\mathbf{n}_i^n, \tag{5.7}$$

where similar notation is used as in (5.5). Here the time step size is set by $\Delta t = 10^{-6}$ and the number of points is $N = 128$. The numerical results are obtained up to $t = 0.2$ for both the star-shaped curve and the complex curve defined in (5.2) and (5.3), respectively. Due to the fourth derivative associated with the term κ_{ss}, Willmore flow

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

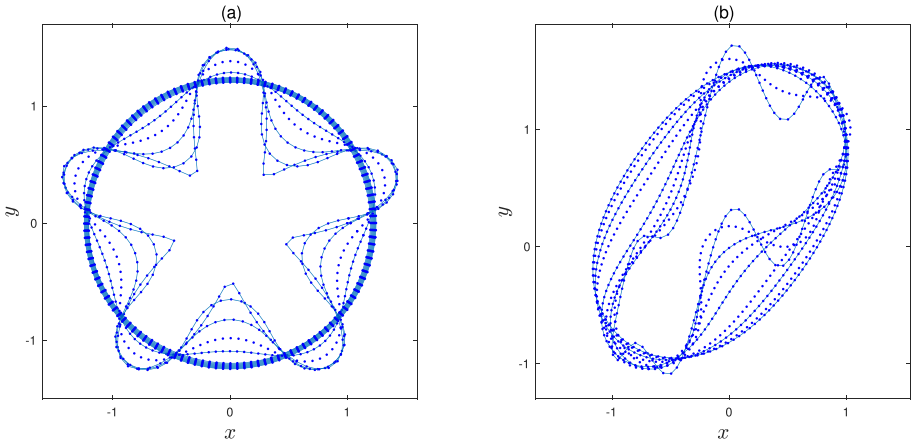

FIG. 15. *For some chosen times, snapshots of the Willmore flow of two curves approximated with* 128 *points:* (a) *an initially star-shaped curve defined in* (5.2), (b) *an initially complex curve defined in* (5.3). *Only the equi-arclength redistribution is applied to both curves initially and every time step.*

is much more difficult to solve numerically than mean curvature flow. Fortunately as similarly discussed above and in [11], the equi-arclength condition suppresses the numerical instability induced by the fourth derivative. So, in this test, only the equi-arclength redistribution is applied initially and every time step.

In Figure 15, two different curves, a star-shaped and a complex, are initially given. In Willmore flow, the contribution of the fourth derivative greatly reduces the total squared curvature, so the deformation in a short time is quite large. Nonetheless, in all snapshots we observe that the curves qualitatively satisfy the equi-arclength condition, meaning that our method works well even in the case of large and quick deformation. They deform to a circular curve in a short time, then evolve slowly. In the literature [9, 3], different curves are studied in Willmore flow, where a similar tendency of behaviors is also observed.

5.4. Shear-induced droplet deformation in Stokes flow. As an application to fluid-structure interaction problems, we consider a droplet immersed in Stokes flow. The evolving motion of a droplet is governed by

$$0 = -\nabla p + \mu \Delta \mathbf{u} - \int_0^{2\pi} \sigma \kappa \mathbf{n} \delta(\mathbf{x} - \mathbf{X}(\theta)) \|\mathbf{X}_\theta\| \mathrm{d}\theta,$$

$$\nabla \cdot \mathbf{u} = 0,$$

$$\frac{\partial \mathbf{X}}{\partial t} = \mathbf{u},$$

where $\mathbf{x}$ is the Eulerian points in bulk fluid, $\mathbf{X}$ is the Lagrangian points on a droplet, and $\delta(\cdot)$ is the Dirac delta function. We denoted the fluid pressure by p, the fluid velocity by $\mathbf{u}$, the fluid viscosity by μ, and the surface tension of a droplet by σ. As studied in [6], for a blob given by $\phi_\epsilon(\mathbf{X}) = 2\epsilon^3/[2\pi(\|\mathbf{X}\|^2 + \epsilon^2)^{5/2}]$, the regularized formulation of fluid velocity in two-dimensional Stokes flow is

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

$$\mathbf{u}(\theta_i,t)=\mathbf{u}_0(\theta_i,t)+\sum_{k=1}^{N}\frac{-\mathbf{f}_k}{4\pi\mu}\left[\log(R_k+\epsilon)-\frac{\epsilon(R_k+2\epsilon)}{(R_k+\epsilon)R_k}\right]+\frac{1}{4\pi\mu}[\mathbf{f}\cdot(\mathbf{X}_i-\mathbf{X}_k)](\mathbf{X}_i-\mathbf{X}_k)\frac{R_k+2\epsilon}{(R_k+\epsilon)^2R_k}, \tag{5.8}$$

where $\mathbf{X}_i=\mathbf{X}(\theta_i)$ and $R_k=\sqrt{\|\mathbf{X}_i-\mathbf{X}_k\|^2+\epsilon^2}$. The position vector is updated accordingly by

$$\frac{\mathbf{X}_i^{n+1}-\mathbf{X}_i^n}{\Delta t}=\mathbf{u}(\theta_i,t). \tag{5.9}$$

The background shear flow is given by $\mathbf{u}_0(\theta_i,t)=(Y(\theta_i,t),0)$ for the droplet initially configured as the unit circle. The blob size denoted by ϵ must be chosen carefully so that the volume loss of the droplet is small enough during simulations. In this test, we set it by $\epsilon=0.2\|\mathbf{X}_\theta(\theta_k)\|\Delta\theta$ adjusting based on the magnitude of arclength element at each point. The fluid viscosity is set by $\mu=1$ and the time step size is by $\Delta t=10^{-4}$. The numerical solution is obtained up to $t=5$.

In Figure 16, droplet motions are plotted, where different surface tensions are set by $\sigma=2.5$ in (a), (b) and $\sigma=1.25$ in (c), (d). When $\sigma=2.5$, the droplet

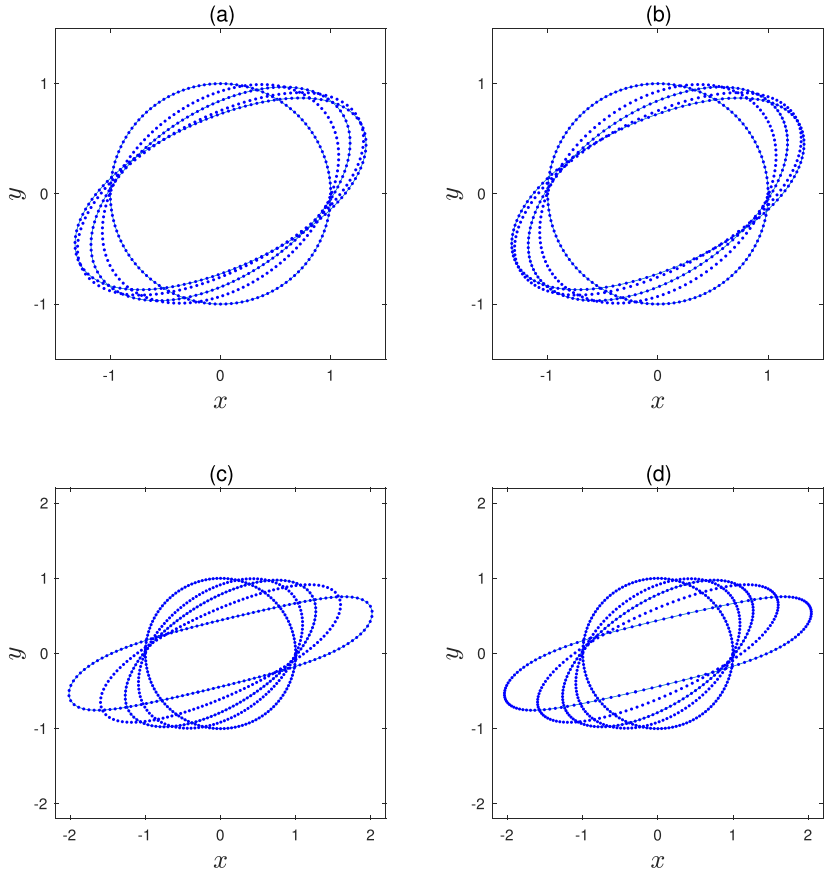

FIG. 16. *For some chosen times, under shear flow, snapshots of initially circular droplet deformation with* $\sigma=2.5$ *in* (a), (b) *and* $\sigma=1.25$ *in* (c), (d)*:* (a), (c) *equi-arclength redistribution,* (b), (d) *curvature-dependent redistribution. Both redistributions are applied initially and every* 1000 *time steps.*

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

deforms to an elliptical shape, then tends to tank-tread. In this case, the difference of effects between two redistributions is ignorable. Contrary to this, when $\sigma = 1.25$, the droplet elongates more, and the effect of curvature-dependent redistribution is clearly shown at the tips of the droplet. In droplet dynamics, the region with high curvature is important since its approximation with more points captures the droplet motion better. Our numerical results indicate that the curvature-dependent redistribution has great potential to be applied in practice.

When the surface tension decreases more to $\sigma = 0.625$, the droplet in shear flow tends to elongate more. Figure 17 shows the higher elongation under both types of remeshing. As seen, in both cases, the initial 128 points increases as time goes on and becomes about 270 points in the final time. One can certainly see that two types of remeshing still satisfy their own condition very well. In the case of the equi-arclength one (a), we observe almost constant distance between two consecutive points as we defined the desired number of points by (3.10), whereas better approximation of droplet tips is shown in the case of the curvature-dependent one (b).

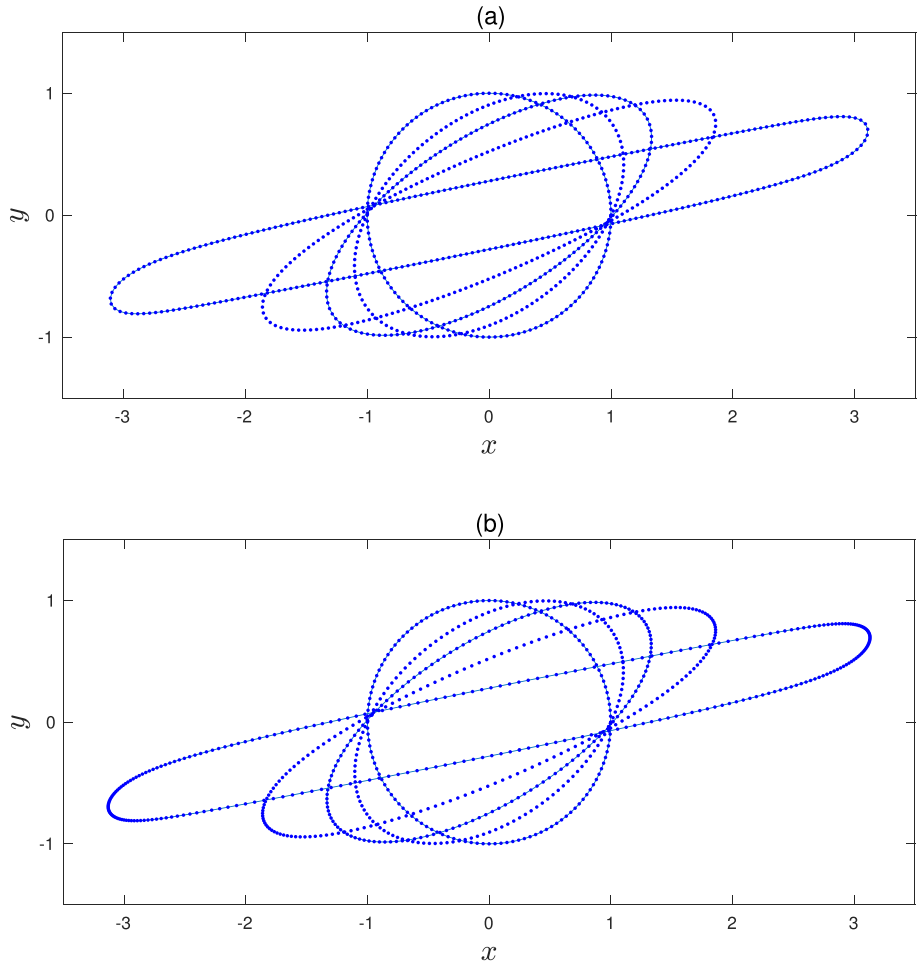

FIG. 17. *By applying remeshing, for some chosen times, under shear flow, snapshots of initially circular droplet deformation with* $\sigma = 0.625$*:* (a) *equi-arclength remeshing,* (b) *curvature-dependent remeshing. The remehsing is applied initially and every* 1000 *time steps.*

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

6. Conclusions and future work. In this paper we presented a numerical method to obtain the equi-arclength(-like) redistribution of points on evolving closed curves. In the framework of the Fourier representation of variables, spectral accuracy was obtained for smoothly perturbed points on the unit circle. In addition, our approach was able to obtain convergent results not only for nonsmoothly perturbed points, but also in the sense of statistics for randomly perturbed points. For other various curves, the equi-arclength redistribution and the curvature-dependent redistribution performed well, still achieving spectral accuracy, although a larger number of points was required to increase the accuracy for a complex curve, particularly with the curvature-dependent one. To check the applicability of our method, we considered three flows: mean curvature flow, Willmore flow, and Stokes flow. In all cases, both types of redistribution were successfully applied. As discussed here and in the literature, the equi-arclength redistribution was stable regardless of its application period and the number of points used. Unlike this, the curvature-dependent redistribution is slightly unstable when it is often used and when few points are used. In practice, however, the curvature-dependent redistribution had great potential, giving qualitatively better approximation in the presence of high curvature regions such as elongated droplet tips. We further showed that our algorithm is applicable to a periodic planar curve without any modification. Although not shown here, our method is not limited to a curve in two dimensions, so one can further apply it to a curve in arbitrary dimensions by defining proper functions associated with arclength and curvature.

Our method can be useful in time-dependent dynamics coupled with a scalar function. For instance, an insoluble surfactant as a scalar function flows along the curve and needs to be interpolated according to the points redistribution. Our method enables one to spectrally interpolate the mass density of surfactant using the newly obtained angle. Meanwhile, since a doubly periodic surface consists of periodic curves in two directions, our approach may be adapted to such surfaces. Fast computing of Fourier series over nonequispaced angles as in [8] will be helpful to speed up our algorithm, and theoretical studies on convergence analysis as in [21, 25] are also promising. In addition, application of our method to an open curve is an obvious extension. We leave these for future work.

REFERENCES

[1] H. W. Bai, Z. L. Ye, and M. Shi, *An improved approximate arc-length parameterization method for Bezier curves*, in Proceedings of the 7th International Conference on Computer-Aided Industrial Design and Conceptual Design, Hangzhou, China, 2006, pp. 699–702.

[2] G. R. Baker and M. J. Shelley, *On the connection between thin vortex layers and vortex sheets*, J. Fluid Mech., 215 (1990), pp. 161–194.

[3] 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.

[4] D. Beylkin and V. Rokhlin, *Fitting a bandlimited curve to points in a plane*, SIAM J. Sci. Comput., 36 (2014), pp. A1048–A1070.

[5] M. G. Blyth, E. I. Părău, and J.-M. Vanden-Broeck, *Hydroelastic waves on fluid sheets*, J. Fluid Mech., 689 (2011), pp. 541–551.

[6] R. Cortez, *The method of regularized Stokeslets*, SIAM J. Sci. Comput, 23 (2001), pp. 1204–1225.

[7] C. de Boor, *Good approximation by splines with variable knots*, in Spline Functions and Approximation Theory, A. Meir and A. Sharma, eds., Birkhäuser Verlag, Basel, 1973, pp. 57–73.

[8] A. Dutt and V. Rokhlin, *Fast Fourier transforms for nonequispaced data*, SIAM J. Sci. Comput., 14 (1993), pp. 1368–1393.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[9] G. Dziuk, E. Kuwert, and R. Schätzle, *Evolution of elastic curves in $\mathbb{R}^n$: Existence and computation*, SIAM J. Math. Anal., 33 (2002), pp. 1228–1245.
[10] S. Fukushima and K. Namikawa, *Generation of a closed curve with Fourier representation*, Systems Computers Japan, 19 (1988), pp. 90–97.
[11] T. Y. Hou, J. S. Lowengrub, and M. J. Shelley, *Removing the stiffness from interfacial flows with surface tension*, J. Comput. Phys., 114 (1994), pp. 312–338.
[12] W. Huang and R. D. Russell, *Adaptive Moving Mesh Methods*, Appl. Math. Sci. 174, Springer, New York, 2011.
[13] F. P. Kuhl and C. R. Giardina, *Elliptic Fourier features of a closed contour*, Comput. Graph. Image Proces., 18 (1982), pp. 236–258.
[14] M.-C. Lai, Y.-H. Tseng, and H. Huang, *Numerical simulation of moving contact lines with surfactant by immersed boundary method*, Commun. Comput. Phys., 8 (2010), pp. 735–757.
[15] X. Li and P. Chen, *A parametrization-invariant Fourier approach to planar linkage synthesis for path generation*, Math. Probl. Eng. (2017), pp. 1–16.
[16] V. D. Liseikin, *Grid Generation Methods*, Springer, Berlin, 1999.
[17] 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.
[18] G. R. Marple, P. K. Purohit, and S. Veerapaneni, *Equilibrium shapes of planar elastic membranes*, Phys. Rev. E, 92 (2015), 012405.
[19] K. Mikula and D. Ševčovič, *Evolution of plane curves driven by a nonlinear function of curvature and anisotropy*, SIAM J. Appl. Math., 61 (2001), pp. 1473–1501.
[20] R. S. Millman and G. D. Parker, *Elements of Differential Geometry*, Prentice-Hall, Englewood Cliffs, NJ, 1977.
[21] J. D. Pryce, *On the convergence of iterated remeshing*, IMA J. Numer. Anal., 9 (1989), pp. 315–335.
[22] J. M. Rallison, *The deformation of small viscous drops and bubbles in shear flows*, Ann. Rev. Fluid Mech., 16 (1984), pp. 45–66.
[23] D. P. Simons and S. S. Snibbe, *Arc-Length Reparameterization*, US Patent 6,115,051 A, 2000.
[24] H. Wang, J. Kearney, and K. Atkinson, *Arc-length parameterized spline curves for real-time simulation*, in Proceedings of the 5th International Conference on Curves and Surfaces, 2002, pp. 387–396.
[25] X. Xu, W. Huang, R. D. Russell, and J. F. Williams, *Convergence of de Boor's algorithm for the generation of equidistributing meshes*, IMA J. Numer. Anal., 31 (2011), pp. 580–596.

Downloaded 10/07/20 to 140.113.10.243. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.