



# Sixth-Order Accurate Schemes for Reinitialization and Extrapolation in the Level Set Framework

Tiankui Zhang<sup>1</sup> · Charles W. Wolgemuth<sup>1,2</sup> 

Received: 6 September 2019 / Revised: 19 February 2020 / Accepted: 30 March 2020  
© Springer Science+Business Media, LLC, part of Springer Nature 2020

## Abstract

The level set method is a common approach for handling moving boundary problems, which allows a moving, irregular surface to be described implicitly on a Cartesian grid. This approach often requires reinitialization of the level set function and extrapolation of fields defined only on the interface. Because many applications in physics and engineering involve calculation of second derivatives of the interface curvature and fourth order derivatives of surface fields, accurate simulations of these problems require high-order methods for reinitialization and extrapolation. Here we build off WENO schemes for Hamilton–Jacobi equations to develop novel sixth-order accurate methods for reinitialization and extrapolation. We present numerical results in three dimensional spaces demonstrating fourth-order accuracy of the interfacial curvature and sixth-order accuracy for the extrapolated surface fields. We then show that the extrapolation scheme can be integrated into the closest point method for surface PDEs and present an example of computing geodesic curves on surfaces.

**Keywords** Level set method · Fourth-order accurate curvature · Reinitializaion equation

**Mathematics Subject Classification** 35F21 · 65M06

## 1 Introduction

A wide range of natural phenomena involves the motion of dynamic interfaces. When an elastic ball hits a rigid wall, its shape deforms. A fish undulates in order to swim. The surface of a soap bubble quivers from the air currents that keep it afloat. Beams bend, flags flutter, and seas swell. Even the membranes of the cells that comprise our bodies ruffle,

---

This research was supported by National Institute of Health (NIH) Grant U54CA210172.

---

✉ Charles W. Wolgemuth  
wolg@email.arizona.edu

Tiankui Zhang  
tiankuizhang@email.arizona.edu

<sup>1</sup> Department of Physics, University of Arizona, Tucson, AZ 85721, USA

<sup>2</sup> Department of Molecular and Cellular Biology, University of Arizona, Tucson, AZ 85721, USA

protrude, invaginate, and pinch off. In some cases, the motions of these interfaces involve the dynamics of surface bound components, such as the proteins and lipids that diffuse and react on cell membranes. The level set method, introduced by Osher and Sethian [21], is an extremely simple and elegant numerical framework for these kinds of problems [16,24]. Still, mathematically modeling dynamics in these problems can be challenging when high-order, nonlinear partial differential equations are involved. For instance, the force density for a thin elastic surface with bending rigidity is proportional to the Laplacian of the mean curvature of the surface [13]. Simulation of thin surfaces, such as cell membranes, then requires calculation of second order derivatives of the surface mean curvature, which includes fourth order derivatives of the surface position [13]. The Cahn–Hilliard equation describing phase separation on a surface, as occurs in the formation of lipid rafts [25], leads to fourth order derivatives of an order parameter [10]. To accurately capture these dynamics in the level set framework, convergent methods for fourth order derivatives of the level set function and the surface fields are necessary, which requires the level set function to be locally smooth and the extrapolation scheme to be very accurate. Existing methods do not provide sufficient accuracy to address these types of problems [7,11]. Therefore, we develop a method here that can be used accurately preserve geometric properties, such as high order derivatives of the shape, of dynamic interfaces and can also be used to extrapolate information defined on those surfaces.

In practice, a moving interface is usually represented implicitly as the zero level set  $\Gamma(t)$  of a signed distance function  $\phi(\mathbf{x}, t)$  in the embedding space  $\mathbb{R}^n$  ( $n = 2$  for a curve and  $n = 3$  for a surface), i.e.,  $\Gamma(t) = \{\mathbf{x} \mid \phi(\mathbf{x}, t) = 0, \mathbf{x} \in \mathbb{R}^n\}$ . The dynamics of  $\Gamma(t)$  under a velocity field  $\mathbf{V}$  is then captured by the level set equation [21]:

$$\frac{\partial \phi}{\partial t} + \mathbf{V} \cdot \nabla \phi = 0. \quad (1)$$

Smoothness of the level set function is not guaranteed by Eq. (1). To restore smoothness, Sussman [28] introduced the reinitialization equation:

$$\frac{\partial \phi}{\partial \tau} + \text{Sign}(\phi^0)(|\nabla \phi| - 1) = 0 \quad (2)$$

where  $\phi^0$  is the initial value of  $\phi$  and  $\tau$  is a fictitious time parameter. Eq. (2) restores  $\phi$  to a signed distance function when iterated to equilibrium. To extrapolate surface fields away from the surface, a hyperbolic PDE advecting  $c$  in the normal direction is solved [22]:

$$\frac{\partial c}{\partial \tau} + \text{Sign}(\phi)N \cdot \nabla c = 0 \quad (3)$$

where  $N \equiv \nabla \phi / |\nabla \phi|$  is the normal of the interface and  $c$  represents the scalar field that is being extended away from the surface.

Both Eqs. (2) and (3) are Hamilton–Jacobi equations with initial and boundary conditions

$$\frac{\partial \psi}{\partial \tau} + H(\psi, \nabla \psi) = 0 \quad (4)$$

where  $H$  is the corresponding Hamiltonian. Successful techniques for solving Hamilton–Jacobi equations depend on construction of the numerical Hamiltonian [2], time discretization [26] and space discretization [14,15], and accurate methods have been developed and extensively tested numerically. For spatial discretization, the weighted, essentially-non-oscillatory (WENO) scheme is often used [14]. This method uses divided differences to determine the smoothness of various approximations to the first derivative, and then weights the various approximations to achieve a smooth, and potentially fifth-order accurate, estimate of the

derivative. While this scheme works well when solving the level set equation (Eq. (1)), application of the WENO scheme to the reinitialization equation (Eq. (2)) can yield poor results [14]. The reason for the poor accuracy was pointed out by Russo and Smereka [23], and they developed a second order accurate remedy by modifying the treatment of Eq. (2) near the boundary. Based on the results of Russo and Smereka, Ch  n   and Min [7] obtained a fourth order accurate reinitialization scheme by adopting a cubic ENO interpolation near the boundary and an HJ-WENO scheme away from the boundary.

In this paper, we further develop the ideas in [23] and [7] to design a novel sixth-order accurate scheme for Hamilton–Jacobi equations with boundary condition specified at the zero of a level set function. Note that because Eqs. (2, 3) are hyperbolic, information defined at the zero level set is propagated away from the implicit boundary and an additional boundary condition at the edge of the computational domain is not needed. This paper is organized as follows. In Sect. 2, we briefly introduce numerical discretization of level set-defined geometries and some calculus tools. In Sect. 3, we review WENO schemes of Jiang and Peng for Hamilton–Jacobi equations and present our treatment of boundary nodes in Sect. 3.2. The central idea is to use high order interpolation to locate the exact boundary [7] and interpolate values on the boundary and then calculate WENO derivatives on a nonuniform stencil near boundary nodes. In Sect. 4, we present numerical results of our scheme. First, we show that we can solve the reinitialization equation (Eq. (2)) with sixth-order accuracy and interface curvature can be computed with fourth-order accuracy, which then provides a convergent scheme for calculating bending forces of soft objects that depend on second order derivatives of interface curvature. Next, we show that Eq. (3) can be solved with sixth-order accuracy and we are able to compute BiLaplacian of a surface field with second-order accuracy. Finally, we show that the extrapolation procedure can be integrated into the closet point method [19] and give an example for computing geodesics on curved surfaces.

## 2 Discretization of Geometries in the Level Set Framework

A strength of the level set formalism is that it allows irregular surfaces to be defined implicitly on a regular grid, such as a simple, rectangular Cartesian grid. Computing geometric properties of an implicit interface represented by the level set function  $\phi(x, y, z)$  is then much easier than that for a triangulated surface. Most of the geometrical information about the interface is encoded in the gradient  $\nabla\phi$  and Hessian  $H(\phi)$  of  $\phi$ , which in Cartesian coordinates are defined as

$$\nabla\phi = (\phi_x \quad \phi_y \quad \phi_z) \tag{5}$$

$$\text{Hessian}(\phi) = \begin{pmatrix} \phi_{xx} & \phi_{xy} & \phi_{xz} \\ \phi_{yx} & \phi_{yy} & \phi_{yz} \\ \phi_{zx} & \phi_{zy} & \phi_{zz} \end{pmatrix} \tag{6}$$

For example, the interface normal  $N$  and mean curvature  $K_M$  are [9]

$$N = \frac{\nabla\phi}{|\nabla\phi|}$$

and

$$K_M = \frac{-|\nabla\phi|^2 \text{Trace}(\text{Hessian}) + \nabla\phi \otimes \text{Hessian}(\phi) \otimes (\nabla\phi)^T}{|\nabla\phi|^3} \tag{7}$$

where we used the convention that the mean curvature for a unit sphere is  $-2$ . The interface Gaussian curvature  $K$  is [9]

$$K = \frac{\nabla\phi \otimes \text{Hessian}^*(\phi) \otimes (\nabla\phi)^T}{|\nabla\phi|^4} \tag{8}$$

where  $\text{Hessian}^*(\phi)$  is the cofactor matrix of the Hessian matrix. The Laplace–Beltrami operator  $\Delta_{\parallel}$  on this interface can be defined using

$$\Delta_{\parallel} = (\nabla - NN \cdot \nabla) \cdot (\nabla - NN \cdot \nabla)$$

Fields distributed in space can be localized to the interface using the Delta function  $\delta(\phi)$  supported on a level set. In order to define  $\delta(\phi)$ , we follow [30] and define  $I(\phi) = \int_0^\phi H(\phi')d\phi' = \max(\phi, 0)$  on the grid. Then the numerical Heaviside function  $H(\phi)$  and  $\delta(\phi)$  are computed by

$$H(\phi) = \frac{\nabla I \cdot \nabla\phi}{|\nabla\phi|^2} \tag{9}$$

$$\delta(\phi) = \frac{\Delta I(\phi) - H(\phi)\Delta\phi}{|\nabla\phi|^2} \tag{10}$$

The integral of a scalar field  $f$  over the interface is then given by a volume integral

$$\int f dA = \int f\delta(\phi)|\nabla\phi|dV. \tag{11}$$

In [3], the level set method is extended to objects with codimension more than one. In particular, we can use an auxiliary level set function  $\psi(x, y, z)$  to represent a curve on the interface. This curve will be the intersection of the zero level sets of both  $\phi$  and  $\psi$ . The unit tangent of this embedded curve is

$$\mathbf{t} = \frac{\nabla\phi \times \nabla\psi}{|\nabla\phi \times \nabla\psi|}. \tag{12}$$

A unit normal for this curve is defined by

$$\mathbf{n} = \mathbf{t} \times \mathbf{N}, \tag{13}$$

and the geodesic curvature of the curve is

$$k_g = \mathbf{n} \cdot \{(\mathbf{t} \cdot \nabla)\mathbf{t}\} \tag{14}$$

### 3 Spatial and Time Discretizations for Hamilton–Jacobi Equations

Hamilton–Jacobi equations (Eq. (4)) involve directionality in the flow of information that carries the function  $\psi$  at one time to its values at later times. As such, it is crucial to use an upwind spatial discretization along with an accurate time stepping routine. In this section, we describe the standard spatial and temporal discretization schemes that are used with the level set method and then provide a detailed description of our modification to the ENO scheme for non-uniform grids that was originally developed in [7].

All of our computations are carried out on three dimensional Cartesian grids, with the level set function and its geometry fields (normals, curvatures etc.) defined at the nodes of the grid.

Fourth order finite difference schemes are used to compute derivatives in Eqs. (5–14). For instance [4],

$$\begin{aligned} \phi_x &= \frac{1}{12\Delta x}(-\phi_{i+2,j,k} + 8\phi_{i+1,j,k} - 8\phi_{i-1,j,k} + \phi_{i-2,j,k}) \\ \phi_{xx} &= \frac{1}{12(\Delta x)^2}(-\phi_{i+2,j,k} + 16\phi_{i+1,j,k} - 30\phi_{i,j,k} + 16\phi_{i-1,j,k} - \phi_{i-2,j,k}) \\ \phi_{xy} &= \frac{1}{48\Delta x\Delta y} \left( -\phi_{i+2,j+2,k} + 16\phi_{i+1,j+1,k} + \phi_{i-2,j+2,k} - 16\phi_{i-1,j+1,k} \right. \\ &\quad \left. + \phi_{i+2,j-2,k} - 16\phi_{i+1,j-1,k} - \phi_{i-2,j-2,k} + 16\phi_{i-1,j-1,k} \right), \end{aligned}$$

with similar constructions for the derivatives along the other directions.

### 3.1 Spatial Discretization

Equation (4) can be written in a semi-discrete form as

$$\frac{\partial \psi}{\partial \tau} + \hat{H}(D_x^- \psi, D_x^+ \psi; D_y^- \psi, D_y^+ \psi; D_z^- \psi, D_z^+ \psi) = 0 \tag{15}$$

where  $D_x^\pm \psi, D_y^\pm \psi, D_z^\pm \psi$  are the one-sided derivatives of  $\psi$  and  $\hat{H}(D_x^- \psi, D_x^+ \psi; \dots)$  is a numerical approximation of  $H(\psi, \nabla \psi)$ . Among all monotone schemes to construct  $\hat{H}$ , Godunov schemes introduce the least numerical diffusion and will be used for our solution. Godunov schemes, however, can be difficult to implement numerically for a general Hamiltonian, with specific Hamiltonians requiring their own specific treatment. For the reinitialization equation (Eq. (2)),

$$H(\phi, \nabla \phi) = \text{Sign}(\phi^0)(|\nabla \phi| - 1)$$

and

$$\hat{H}(D_x^- \psi, D_x^+ \psi; D_y^- \psi, D_y^+ \psi; D_z^- \psi, D_z^+ \psi) = \text{Sign}(\phi^0) \left\{ \sqrt{\phi_x^2 + \phi_y^2 + \phi_z^2} - 1 \right\} \tag{16}$$

where

$$\phi_x^2 \equiv \max(\max(D_x^- \phi, 0)^2, \min(D_x^+ \phi, 0)^2)$$

when  $\text{Sign}(\phi^0) > 0$  and

$$\phi_x^2 \equiv \max(\min(D_x^- \phi, 0)^2, \max(D_x^+ \phi, 0)^2)$$

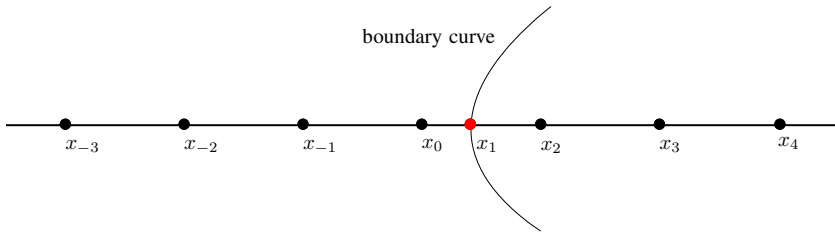
when  $\text{Sign}(\phi^0) \leq 0$ .  $\phi_y^2$  and  $\phi_z^2$  are defined similarly. For the extrapolation equation (Eq. (3)),

$$H(\nabla c) = \text{Sign}(\phi) N \cdot \nabla c$$

and

$$\hat{H}(D_x^- c, D_x^+ c; D_y^- c, D_y^+ c; D_z^- c, D_z^+ c) = \sum_{i=x,y,z} \{ \min(V_i, 0) D_i^+ c + \max(V_i, 0) D_i^- c \}. \tag{17}$$

where we define  $V \equiv \text{Sign}(\phi)N$ . Numerically,  $\text{Sign}(\phi)$  can take a value among 1, -1 and 0. We set  $\text{Sign}(\phi)$  to be zero when  $\text{abs}(\phi) < 10^{-6} \times \Delta x$ , where  $\Delta x$  is the grid spacing.



**Fig. 1** When the boundary passes between nodes  $x_0$  and  $x_2$ , an accurate estimate for the off-grid location of the boundary  $x_1$  is used when calculating the one-sided WENO derivatives for  $x_0$  and  $x_2$

### 3.2 One-Sided WENO Derivatives on a Nonuniform Grid

Solving the reinitialization equation (Eq. (2)) or extension equation (Eq. (3)), requires forward and backward one-sided derivatives. For nodes not immediately next to the boundary, the standard WENO schemes from Jiang [14,15] can be applied to calculate these derivatives. However, for nodes immediately next to the boundary, special treatments are usually needed [7,23]. In [7], the subcell fix from [23] was augmented using general ENO reconstructions to achieve 4th order accuracy. Here, we utilize the conceptual method for converting from ENO to WENO derivatives in order to further improve the scheme of [7], such that our method can achieve an optimal 6th order accuracy when solving HJ equations involving level set defined boundaries conditions. The key point is to develop WENO schemes for non-uniform grids and then apply this more general WENO scheme to calculate one-sided derivatives for boundary nodes.

#### 3.2.1 Computation of the Location of the Boundary

In three dimensional space, the boundary defined by the zero level set of  $\phi(x, y, z)$  is a two dimensional surface. To compute the one-sided WENO derivatives near the boundary with sufficient accuracy, we need to locate the points where the zero contour crosses between grid nodes. We define a grid node at point  $(x_0, y_0, z_0)$  to be a boundary node if  $\phi(x_0, y_0, z_0)$  differs in sign from any of its six nearest neighbors on the Cartesian grid.

For instance, if  $\phi(x_0, y_0, z_0)\phi(x_0 + \Delta x, y_0, z_0) < 0$ , both  $(x_0, y_0, z_0)$  and  $(x_0 + \Delta x, y_0, z_0)$  are boundary nodes. To compute  $D_x^\pm \phi$  at  $(x_0, y_0, z_0)$  and  $(x_0 + \Delta x, y_0, z_0)$ , we begin by localizing the crossing between the boundary and the line segment  $(x_0 + \xi \Delta x, y_0, z_0)$ ,  $\xi \in (0, 1)$ . This is illustrated in Fig. 1, where the boundary passes between  $x_0$  and  $x_2$  at the location defined as  $x_1$ . Note that all the nodes shown are grid nodes except for  $x_1$ . For this calculation, we are only concerned with the crossing point along the  $x$  direction. Crossing points along the  $y$  and  $z$  directions are handled in an identical manner.

When  $\phi(x_0)\phi(x_{-1}) < 0$  or  $\phi(x_2)\phi(x_3) < 0$ , there is a kink near  $x_0$  or  $x_2$ . We then use  $(x_{-1}, \phi_{-1}), (x_0, \phi_0), (x_2, \phi_2), (x_3, \phi_3)$  to construct a quadratic ENO polynomial of  $\phi$  and the root  $x_1$  of this polynomial approximates the boundary location [20]. In particular [20],

$$x_1 = \begin{cases} x_0 + \Delta x \left( \frac{1}{2} + \frac{\phi_0 - \phi_2 - \text{sign}(\phi_0 - \phi_2)\sqrt{D}}{\phi_{xx}^0} \right) & \text{if } |\phi_{xx}^0| > 10^{-10} \\ x_0 + \Delta x \frac{\phi^0}{\phi^0 - \phi^2} & \text{else} \end{cases}$$

where  $\phi_{xx}^0 = \text{minmod}(\phi_{-1} - 2\phi_0 + \phi_2, \phi_0 - 2\phi_2 + \phi_3)$ ,  $D = (\phi_{xx}^0/2 - \phi_0 - \phi_2)^2 - 4\phi_0\phi_2$  and the minmod function is defined as

$$\text{minmod}(\alpha, \beta) = \begin{cases} 0 & \text{if } \alpha\beta \leq 0 \\ \alpha & \text{if } \alpha\beta > 0 \text{ and } |\alpha| \leq |\beta| \\ \beta & \text{if } \alpha\beta > 0 \text{ and } |\alpha| > |\beta| \end{cases}$$

When a kink is not present near  $x_0$  and  $x_2$ , we construct a 5th order Lagrange polynomial that interpolates through  $(x_i, \phi_i)$ ,  $i \in \{-2, -1, 0, 2, 3, 4\}$  and use Brent’s method to find the root  $x_1$  that lies between  $x_0$  and  $x_2$ . Then  $(x_1, \phi_1 = 0)$  is included in the stencil to construct  $D_x^\pm \phi$  at  $x_0$  and  $x_2$ . For later use, we define  $\Delta x^+ = x_1 - x_0$ , which is the distance between node  $(x_0, y_0, z_0)$  and the boundary along the positive  $x$  direction. If the boundary is to the left of  $x_0$  in Fig. 1, this distance is then denoted as  $\Delta x^-$ , which will be the distance between node  $(x_0, y_0, z_0)$  and the boundary along the negative  $x$  direction. In a similar way, we can define  $\Delta y^\pm$  and  $\Delta z^\pm$  for the boundary nodes.

For an advected field  $\psi$  other than the level set function  $\phi$ , a Lagrange polynomial  $l_5(x)$  interpolating through  $(x_i, \psi_i)$ ,  $i \in \{-2, -1, 0, 2, 3, 4\}$  is constructed and  $(x_1, l_5(x_1))$  will be included in the stencil to construct  $D_x^\pm \psi$  at  $x_0$  and  $x_2$ .

In our analysis, we assumed that the zero level set is sufficiently far off from the boundary of the grid mesh so that we can always find enough grid points near the boundary to do the interpolation mentioned above. The accuracy of the boundary location procedure depends on the smoothness of the boundary and the advected field. Near a kink, the boundary location can only be determined with first order accuracy and the algorithm is then only first order accurate [20].

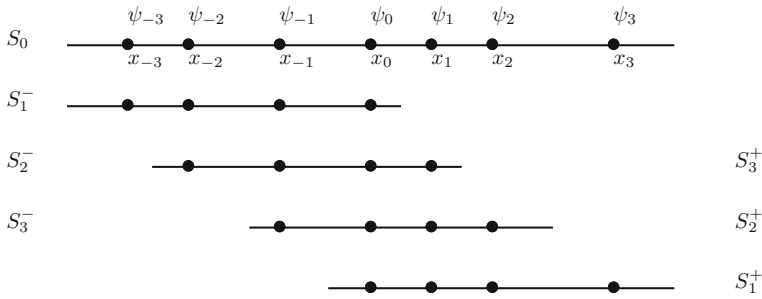
The computation of boundary locations and the interpolation of other advected fields is carried out in a dimension by dimension manner. This process applies in the same way, regardless of the dimensionality of the problem.

### 3.2.2 Computation of WENO Derivatives

To compute the one-sided derivatives at node  $x_0$ , we begin by defining a seven-point stencil about this node (Fig. 2). Note that when  $x_0$  is a boundary node as is shown in Fig. 1, the node  $(x_1, \psi_1)$  on the boundary should be included in the stencil, in which case the stencil will be nonuniform. In all other cases, the stencil is uniform. We maintain full generality by allowing all of the nodes to be non-uniformly spaced.

To compute  $D_x^- \psi(x_0)$ , a left biased stencil  $S^- = \{(x_i, \psi_i) | i \in \{-3, -2, -1, 0, 1, 2\}\}$  should be used. The first step is to break  $S^-$  into three candidate ENO (essentially non-oscillatory) stencils  $S_i^-$ ,  $i = 1, 2, 3$  and approximate  $\psi(x)$  on those stencils with polynomial interpolations  $p_i^-(x)$ ,  $i = 1, 2, 3$ . For instance,  $p_2^-(x)$  will be a third order polynomial interpolating through all points in  $S_2^- = \{(x_i, \psi_i) | i \in \{-2, -1, 0, 1\}\}$  and  $u_2^- \equiv \frac{dp_2^-(x)}{dx}|_{x=x_0}$  will be a candidate for  $D_x^- \psi(x_0)$  in the ENO scheme. It is straightforward to compute all  $u_i^-$ ,  $i = 1, 2, 3$  and the results are [27]

$$\begin{aligned} u_1^- &= \bar{u}_{-\frac{3}{2}} + \left(\frac{x_0 - x_{-1}}{x_{-1} - x_{-3}}\right) \left(\frac{x_0 - x_{-2}}{x_0 - x_{-3}}\right) (\bar{u}_{-\frac{5}{2}} - \bar{u}_{-\frac{3}{2}}) \\ &\quad + \left(1 + \frac{x_0 - x_{-1}}{x_0 - x_{-2}} + \frac{x_0 - x_{-1}}{x_0 - x_{-3}}\right) (\bar{u}_{-\frac{1}{2}} - \bar{u}_{-\frac{3}{2}}) \\ u_2^- &= \bar{u}_{-\frac{1}{2}} + \left(\frac{x_0 - x_{-1}}{x_1 - x_{-2}}\right) \left(\frac{x_0 - x_{-2}}{x_1 - x_{-1}}\right) (\bar{u}_{\frac{1}{2}} - \bar{u}_{-\frac{1}{2}}) \end{aligned}$$



**Fig. 2** A general, nonuniform, seven-point stencil,  $S_0$ , is used to find backward and forward derivatives at  $x_0$ . In the WENO scheme, the full stencil is broken into substencils ( $S_j^\pm$ ), and the derivative is approximated on each of these substencils. A weighted average of these approximations is then used to define the forward and backward WENO derivatives

$$\begin{aligned}
 u_3^- = & \bar{u}_{\frac{1}{2}} - \left( \frac{x_0 - x_{-1}}{x_1 - x_{-2}} \right) \left( \frac{x_1 - x_0}{x_0 - x_{-2}} \right) \left( \bar{u}_{-\frac{3}{2}} - \bar{u}_{-\frac{1}{2}} \right) \\
 & + \left( \frac{x_1 - x_0}{x_2 - x_{-1}} \right) \left( \frac{x_2 - x_0}{x_1 - x_{-1}} \right) \left( \bar{u}_{-\frac{1}{2}} - \bar{u}_{\frac{1}{2}} \right) \\
 & - \left( \frac{x_1 - x_0}{x_2 - x_{-1}} \right) \left( \frac{x_0 - x_{-1}}{x_2 - x_0} \right) \left( \bar{u}_{\frac{3}{2}} - \bar{u}_{\frac{1}{2}} \right)
 \end{aligned} \tag{18}$$

where  $\bar{u}_{r+\frac{1}{2}}, r \in \{-3, -2, -1, 0, 1, 1\}$  are first Newton divided differences defined as

$$\bar{u}_{r+\frac{1}{2}} = \frac{\psi_{r+1} - \psi_r}{x_{r+1} - x_r}.$$

The expressions for  $u_j^+$  can be obtained by the following reflection transformation:

$$x_r \rightarrow x_{-r}, \bar{u}_{r+\frac{1}{2}} \rightarrow \bar{u}_{-(r+\frac{1}{2})}. \tag{19}$$

In what follows, we will only list formula with superscript  $-$  since those with superscript  $+$  can be obtained from reflection using Eq. (19).

In the ENO schemes introduced by Harten and Osher [12],  $D_x^- \psi(x_0)$  is approximated by the  $u_i^-$  from Eq. (18), which corresponds to the substencil  $S_i^-$  where  $\psi(x)$  varies most smoothly. Consequently, ENO schemes approximate  $D_x^- \psi(x_0)$  with only third order accuracy on a six point stencil  $S^-$ , since information from the other substencils are not used. WENO schemes use the same substencil approximations to the first derivatives,  $u_1^-, u_2^-, u_3^-$ , but employ a convex combination  $\sum_{i=1}^3 \omega_i^- u_i^-$  of the substencil derivatives to approximate  $D_x^- \psi(x_0)$ , thereby increasing the accuracy up to potentially 5th order [18].

When  $\psi(x)$  is smooth over all stencils, the weights  $\omega_i^-$  should approximately cancel truncation errors in  $u_i^-$  to achieve optimal accuracy. Let us denote by  $C_i^-$  weights that will give the optimal fifth order accuracy for  $D_x^- \psi(x_0)$ . In other words, if a fifth order polynomial  $p^-(x)$  is constructed to interpolate through all points in  $S^-$ , the optimal weights  $C_i^-$  should satisfy

$$\frac{dp^-(x)}{dx} \Big|_{x=x_0} = C_1^- u_1^- + C_2^- u_2^- + C_3^- u_3^-,$$



which gives uniquely [27]

$$\begin{aligned}
 C_1^- &= \left(\frac{x_1 - x_0}{x_2 - x_{-3}}\right) \left(\frac{x_2 - x_0}{x_1 - x_{-3}}\right) \\
 C_2^- &= \left(\frac{x_0 - x_{-3}}{x_2 - x_{-3}}\right) \left(\frac{x_2 - x_0}{x_2 - x_{-2}} + \frac{x_2 - x_0}{x_1 - x_{-3}}\right) \\
 C_3^- &= \left(\frac{x_0 - x_{-3}}{x_2 - x_{-3}}\right) \left(\frac{x_0 - x_{-2}}{x_2 - x_{-2}}\right).
 \end{aligned}
 \tag{20}$$

When  $\psi(x)$  is not smooth over the full stencil, non-smooth substencils should be given smaller weights. The smoothness indicator  $IS_i^-$  for stencil  $S_i^-$  is a weighted measure of the integrated square of the 2nd and 3rd derivatives over the substencil, which is given by Jiang and Shu [15],

$$IS_i^- = \sum_{l=1}^2 \int_{x_{-1}}^{x_0} (x_0 - x_{-1})^{2l-1} \left(\frac{d^{l+1} p_i^-(x)}{d^{l+1} x}\right)^2 dx,$$

leading to [27]

$$\begin{aligned}
 IS_1^- &= 40(x_0 - x_{-1})^4 (\bar{w}_{-\frac{3}{2}})^2 \\
 &\quad + 4 \left(\frac{x_0 - x_{-1}}{x_0 - x_{-3}}\right)^2 \left( \{(x_0 - x_{-2})\bar{v}_{-2} - (2x_0 - x_{-2} - x_{-3})\bar{v}_{-1}\} \right. \\
 &\quad \left. \times \{(x_{-1} - x_{-2})\bar{v}_{-2} - (x_0 + x_{-1} - x_{-2} - x_{-3})\bar{v}_{-1}\} \right) \\
 IS_2^- &= 40(x_0 - x_{-1})^4 (\bar{w}_{-\frac{1}{2}})^2 + 4 \left(\frac{x_0 - x_{-1}}{x_1 - x_{-2}}\right)^2 \left( \{(x_0 - x_1)\bar{v}_{-1} - (x_0 - x_{-2})\bar{v}_0\} \right. \\
 &\quad \left. \times \{(x_{-1} - x_1)\bar{v}_{-1} - (x_{-1} - x_{-2})\bar{v}_0\} \right) \\
 IS_3^- &= 40(x_0 - x_{-1})^4 (\bar{w}_{\frac{1}{2}})^2 \\
 &\quad + 4 \left(\frac{x_0 - x_{-1}}{x_2 - x_{-1}}\right)^2 \left( \{(2x_{-1} - x_1 - x_2)\bar{v}_0 - (x_{-1} - x_1)\bar{v}_1\} \right. \\
 &\quad \left. \times \{(x_0 + x_{-1} - x_1 - x_2)\bar{v}_0 - (x_0 - x_1)\bar{v}_1\} \right),
 \end{aligned}
 \tag{21}$$

where  $\bar{v}_i$  and  $\bar{w}_i$  are second and third Newton divided differences defined by

$$\bar{v}_i \equiv \frac{1}{x_{i+1} - x_{i-1}} (\bar{u}_{i+\frac{1}{2}} - \bar{u}_{i-\frac{1}{2}}), \quad \bar{w}_{i+\frac{1}{2}} \equiv \frac{1}{x_{i+2} - x_{i-1}} (\bar{v}_{i+1} - \bar{v}_i).$$

Weights that approximate the optimal weights  $C_i^-$  in smooth regions while suppressing oscillations in non-smooth regions are then defined by Liu et al. [18]

$$\omega_j^- = \frac{\alpha_j^-}{\sum_{k=1}^3 \alpha_k^-}, \quad \alpha_j^- = \frac{C_j^-}{(\epsilon + IS_j^-)^2}, \quad \epsilon = 10^{-6}.$$

The WENO derivatives  $D_x^- \psi(x_0)$  are then defined as

$$D_x^- \psi(x_0) = \sum_{j=1}^3 \omega_j^- u_j^-.$$

When the grid is uniform, Eqs. (18, 20, 21) reduce to

$$\begin{aligned}
 u_1^- &= \frac{1}{3}\bar{u}_{-\frac{5}{2}} - \frac{7}{6}\bar{u}_{-\frac{3}{2}} + \frac{11}{6}\bar{u}_{-\frac{1}{2}}, C_1^- = 0.1 \\
 u_2^- &= -\frac{1}{6}\bar{u}_{-\frac{3}{2}} + \frac{5}{6}\bar{u}_{-\frac{1}{2}} + \frac{1}{3}\bar{u}_{\frac{1}{2}}, C_2^- = 0.6 \\
 u_3^- &= \frac{1}{3}\bar{u}_{-\frac{1}{2}} + \frac{5}{6}\bar{u}_{\frac{1}{2}} - \frac{1}{6}\bar{u}_{\frac{3}{2}}, C_3^- = 0.3
 \end{aligned}$$

and

$$\begin{aligned}
 IS_1^- &= \frac{13}{12} \left( \bar{u}_{-\frac{5}{2}} - 2\bar{u}_{-\frac{3}{2}} + \bar{u}_{-\frac{1}{2}} \right)^2 + \frac{1}{4} \left( \bar{u}_{-\frac{5}{2}} - 4\bar{u}_{-\frac{3}{2}} + 3\bar{u}_{-\frac{1}{2}} \right)^2 \\
 IS_2^- &= \frac{13}{12} \left( \bar{u}_{-\frac{3}{2}} - 2\bar{u}_{-\frac{1}{2}} + \bar{u}_{\frac{1}{2}} \right)^2 + \frac{1}{4} \left( \bar{u}_{-\frac{3}{2}} - \bar{u}_{\frac{1}{2}} \right)^2 \\
 IS_3^- &= \frac{13}{12} \left( \bar{u}_{-\frac{1}{2}} - 2\bar{u}_{\frac{1}{2}} + \bar{u}_{\frac{3}{2}} \right)^2 + \frac{1}{4} \left( 3\bar{u}_{-\frac{1}{2}} - 4\bar{u}_{\frac{1}{2}} + \bar{u}_{\frac{3}{2}} \right)^2,
 \end{aligned}$$

which are the canonical formulae for the WENO scheme [14]. We refer the reader to [15] for a more thorough discussion of WENO schemes and to [27] for WENO schemes on non-uniform grids.

### 3.3 Time Discretization

A popular time discretization for Eq. (15) is the third order TVD Runge–Kutta scheme [26] with the following Euler steps:

$$\begin{aligned}
 \tilde{\psi}^{n+1} &= \psi^n - \Delta\tau \hat{H} (D_x^- \psi^n, D_x^+ \psi^n; D_y^- \psi^n, D_y^+ \psi^n; D_z^- \psi^n, D_z^+ \psi^n) \\
 \tilde{\psi}^{n+2} &= \tilde{\psi}^{n+1} - \Delta\tau \hat{H} (D_x^- \tilde{\psi}^{n+1}, D_x^+ \tilde{\psi}^{n+1}; D_y^- \tilde{\psi}^{n+1}, D_y^+ \tilde{\psi}^{n+1}; D_z^- \tilde{\psi}^{n+1}, D_z^+ \tilde{\psi}^{n+1}) \\
 \tilde{\psi}^{n+\frac{1}{2}} &= \frac{3}{4}\psi^n + \frac{1}{4}\tilde{\psi}^{n+2} \\
 \tilde{\psi}^{n+\frac{3}{2}} &= \tilde{\psi}^{n+\frac{1}{2}} - \Delta\tau \hat{H} (D_x^- \tilde{\psi}^{n+\frac{1}{2}}, D_x^+ \tilde{\psi}^{n+\frac{1}{2}}; D_y^- \tilde{\psi}^{n+\frac{1}{2}}, D_y^+ \tilde{\psi}^{n+\frac{1}{2}}; D_z^- \tilde{\psi}^{n+\frac{1}{2}}, D_z^+ \tilde{\psi}^{n+\frac{1}{2}}) \\
 \tilde{\psi}^{n+1} &= \frac{1}{3}\psi^n + \frac{2}{3}\tilde{\psi}^{n+\frac{3}{2}}.
 \end{aligned}$$

One benefit of this Runge–Kutta scheme is that relatively large CFL numbers can be used with the WENO scheme. We use 0.3 as our CFL number. In our numerical experiments,  $\Delta\tau$  varies locally and at grid  $(x, y, z)$  is defined by  $\text{Min}$  [20]

$$\Delta\tau = 0.3 \cdot \min(\Delta x^+, \Delta x^-, \Delta y^+, \Delta y^-, \Delta z^+, \Delta z^-),$$

where  $\Delta x^\pm, \Delta y^\pm, \Delta z^\pm$  are taken to be  $\Delta x, \Delta y, \Delta z$  for non-boundary nodes.

## 4 Numerical Results

### 4.1 Computation of Interfacial Curvature and Bending Forces in 3D

A primary goal for developing a high order accurate implementation of the reinitialization equation is to preserve the quality of geometric information stored in the distance map at a sufficient level that the forces derived from the shape remain accurate. In problems

**Table 1** Accuracy results for the reinitialized level set function  $\phi$  for example in Sect. 4.1 after 1000 iterations

	$\ \phi - \phi_h\ _1$	Order	$\ \phi - \phi_h\ _2$	Order	$\ \phi - \phi_h\ _\infty$	Order
$16^3$	$7.122 \times 10^{-6}$	–	$5.283 \times 10^{-6}$	–	$8.013 \times 10^{-6}$	–
$32^3$	$9.772 \times 10^{-8}$	6.19	$6.512 \times 10^{-8}$	6.34	$8.995 \times 10^{-8}$	6.48
$64^3$	$2.761 \times 10^{-9}$	5.15	$1.684 \times 10^{-9}$	5.27	$2.252 \times 10^{-9}$	5.32
$128^3$	$4.384 \times 10^{-11}$	5.98	$2.659 \times 10^{-11}$	5.98	$3.566 \times 10^{-11}$	5.98

**Table 2** Accuracy results for the computation of interface mean curvature  $K_M$ 

	$\ K_M - (K_M)_h\ _1$	Order	$\ K_M - (K_M)_h\ _2$	Order	$\ K_M - (K_M)_h\ _\infty$	Order
$16^3$	$7.100 \times 10^{-3}$	–	$3.754 \times 10^{-3}$	–	$3.139 \times 10^{-3}$	–
$32^3$	$5.066 \times 10^{-4}$	3.81	$3.183 \times 10^{-4}$	3.56	$3.782 \times 10^{-4}$	3.05
$64^3$	$2.432 \times 10^{-5}$	4.38	$1.509 \times 10^{-5}$	4.40	$2.583 \times 10^{-5}$	3.87
$128^3$	$1.510 \times 10^{-6}$	4.01	$9.700 \times 10^{-7}$	3.96	$1.633 \times 10^{-6}$	3.98

**Table 3** Accuracy results for the computation of interface Gaussian curvature  $K$ 

	$\ K - K_h\ _1$	Order	$\ K - K_h\ _2$	Order	$\ K - K_h\ _\infty$	Order
$16^3$	$2.176 \times 10^{-2}$	–	$1.285 \times 10^{-2}$	–	$1.103 \times 10^{-2}$	–
$32^3$	$1.141 \times 10^{-3}$	4.25	$8.449 \times 10^{-4}$	3.93	$1.142 \times 10^{-3}$	3.27
$64^3$	$5.559 \times 10^{-5}$	4.36	$3.722 \times 10^{-5}$	4.50	$7.691 \times 10^{-5}$	3.89
$128^3$	$3.433 \times 10^{-6}$	4.02	$2.451 \times 10^{-6}$	3.92	$4.902 \times 10^{-6}$	3.97

involving elastic surfaces, bending forces depend on the curvatures (mean and Gaussian) and the surface Laplacian of the mean curvature. The simplest form for this force density (up to a multiplication constant) is  $f = \Delta_{\parallel} K_M + K_M^3 - 2K_M K$  [33]. In this section, we test our method's ability to preserve the accuracy of these quantities when an initial level set function (that is not necessarily an exact signed distance map) is reinitialized.

As a first test, we consider a spherical surface defined by an initial level set function  $\phi(x, y, z) = x^2 + y^2 + z^2 - (0.6)^2$  in the  $[-1, 1]^3$  domain. This initial exact distance map is used as the seed for the reinitialization method described in Sect. 2. As shown in Table 1 the reinitialized  $\phi$  maintains sixth-order accuracy. Using the reinitialized  $\phi$ , we then compute the interfacial mean curvature  $K_M$  and Gaussian curvature  $K$  using fourth order accurate approximations to the first and second derivatives, as described in Sec. 5.3.1 of [4]. We compute the  $L_1$  and  $L_2$  errors in these quantities at the interface by integrating the error over the surface using the numerical integration described in Sect. 2. The  $L_\infty$  error is also determined for nodes at the boundary. Tables 2 and 3 demonstrate fourth-order accuracy for the curvatures. Finally, using the same approximations for the derivatives, we compute the surface Laplacian of the mean curvature, which is found to be second-order accurate (Table 4). As a comparison, Table 5 shows accuracy results for  $\Delta_{\parallel} K_M$  using the fourth order accurate scheme from [7]. Comparing these results from Table 5 with Table 9 and Table 12 in [7] shows that much higher accuracy can be achieved on a much smaller grid with our sixth-order accurate scheme.

**Table 4** Accuracy results for the computation of the surface Laplacian of the mean curvature  $\Delta_{\parallel} K_M$ .  $K_M$  field is extended before used to compute  $\Delta_{\parallel} K_M$ .

	$L_1$ -Error	Order	$L_2$ -Error	Order	$L_{\infty}$ -Error	Order
$16^3$	$9.575 \times 10^{-1}$	–	$5.311 \times 10^{-1}$	–	$6.307 \times 10^{-1}$	–
$32^3$	$1.675 \times 10^{-1}$	2.52	$8.952 \times 10^{-2}$	2.57	$8.912 \times 10^{-2}$	2.82
$64^3$	$4.513 \times 10^{-2}$	1.89	$2.898 \times 10^{-2}$	1.63	$5.020 \times 10^{-2}$	0.82
$128^3$	$9.914 \times 10^{-3}$	2.19	$6.825 \times 10^{-3}$	2.09	$1.341 \times 10^{-2}$	1.90

The error is computed as  $\|\Delta_{\parallel} K_M - (\Delta_{\parallel} K_M)_h\|$

**Table 5** Accuracy results for the computation of surface Laplacian of the mean curvature using fourth-order accurate scheme from [7].

	$L_1$ -Error	Order	$L_2$ -Error	Order	$L_{\infty}$ -Error	Order
$16^3$	$3.197 \times 10^0$	–	$1.949 \times 10^0$	–	$1.916 \times 10^0$	–
$32^3$	$3.382 \times 10^0$	–0.08	$1.965 \times 10^0$	–0.01	$1.853 \times 10^0$	0.05
$64^3$	$5.763 \times 10^0$	–0.77	$5.895 \times 10^0$	–1.58	$1.773 \times 10^1$	–3.26
$128^3$	$5.609 \times 10^0$	0.04	$5.574 \times 10^0$	0.08	$2.997 \times 10^1$	–0.76

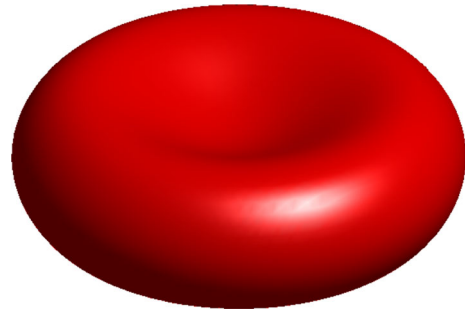
The error is computed as  $\|\Delta_{\parallel} K_M - (\Delta_{\parallel} K_M)_h\|$

As another more complicated example, we consider the red blood cell shape [11] given by the zero level set of

$$\phi(x, y, z) = \left(\frac{2z}{R}\right)^2 - \left(1 - \frac{x^2 + y^2}{R^2}\right) \left(C_0 + C_1 \frac{x^2 + y^2}{R^2} + C_2 \frac{(x^2 + y^2)^2}{R^4}\right)^2 \quad (22)$$

where  $C_0 = 0.2072$ ,  $C_1 = 2.0026$ ,  $C_2 = -1.1228$  and  $R$  is the length of the large half-axis of the RBC and is taken to be 1 here. Note that this function is not an exact signed distance function. The zero level set for this  $\phi$  is shown in Fig. 3. After initializing  $\phi$  according to Eq. (22), we reinitialize it to be a signed distance function and calculate the mean curvature  $K_M$ , Gaussian curvature  $K$  and  $\Delta_{\parallel} K_M$  on the surface. On this more realistic shape, our scheme still produces fourth order accuracy for the computation of interfacial mean and Gaussian curvatures (Tables 6, 7) and can compute  $\Delta_{\parallel} K_M$  with second order accuracy in both the  $L_2$  norm and maximum norm (Tables 8, 9), whereas the scheme from [7] does not provide convergence for the computation of  $\Delta_{\parallel} K_M$ . It is also important to note that schemes that triangulate the surface are not convergent for the computation of  $\Delta_{\parallel} K_M$  in the maximum norm [11]. As previously mentioned, the surface Laplacian of the interface mean curvature is important in the force density of an elastic surface. Many of the existing methods will produce large errors when computing this force. Therefore, high order schemes, such as the one proposed here, are necessary for these problems. We note that a fourth order scheme for the computation of curvature in the level set framework in two dimension has been proposed in [5]. This approach is based on the osculatory circle approximation, which is difficult to implement in three dimensions and fails whenever the curvature is zero. Our approach, however, is straightforward to implement in two or three dimensions, and simple formulae for mean and Gaussian curvatures can be used [9]. Thus our sixth order scheme provides a significant improvement that can enable accurate simulations of three-dimensional soft

**Fig. 3** The red blood cell shape defined by the level set function Eq. (22) (Color figure online)



**Table 6** Accuracy results for the computation of interface mean curvature for the red blood cell shape using our sixth order accurate scheme, where the error is defined as  $\|K_M - (K_M)_h\|$

	$L_1$ -Error	Order	$L_2$ -Error	Order	$L_\infty$ -Error	Order
$16^3$	$1.678 \times 10^0$	–	$7.923 \times 10^{-1}$	–	$8.740 \times 10^{-1}$	–
$32^3$	$2.856 \times 10^{-1}$	2.55	$1.490 \times 10^{-1}$	2.41	$2.511 \times 10^{-1}$	1.80
$64^3$	$1.998 \times 10^{-2}$	3.84	$1.691 \times 10^{-2}$	3.14	$4.728 \times 10^{-2}$	2.41
$128^3$	$1.176 \times 10^{-3}$	4.09	$8.532 \times 10^{-4}$	4.31	$3.506 \times 10^{-3}$	3.75

**Table 7** Accuracy results for the computation of interface Gaussian curvature for the red blood cell shape using our sixth order accurate scheme

	$\ K - K_h\ _1$	Order	$\ K - K_h\ _2$	Order	$\ K - K_h\ _\infty$	Order
$16^3$	$2.657 \times 10^0$	–	$1.266 \times 10^0$	–	$1.296 \times 10^0$	–
$32^3$	$7.583 \times 10^{-1}$	1.81	$5.456 \times 10^{-1}$	1.21	$9.430 \times 10^{-1}$	0.46
$64^3$	$7.225 \times 10^{-2}$	3.39	$8.023 \times 10^{-2}$	2.77	$2.296 \times 10^{-1}$	2.04
$128^3$	$3.924 \times 10^{-3}$	4.20	$3.519 \times 10^{-3}$	4.51	$1.777 \times 10^{-2}$	3.69

**Table 8** Accuracy results for the computation of surface Laplacian of the interface mean curvature for the red blood cell shape using our sixth order accurate scheme.

	$L_1$ -Error	Order	$L_2$ -Error	Order	$L_\infty$ -Error	Order
$16^3$	$2.549 \times 10^2$	–	$1.094 \times 10^2$	–	$9.351 \times 10^2$	–
$32^3$	$1.680 \times 10^2$	0.60	$7.791 \times 10^1$	0.49	$8.049 \times 10^1$	0.22
$64^3$	$2.884 \times 10^1$	2.54	$2.133 \times 10^1$	1.87	$6.434 \times 10^1$	0.32
$128^3$	$4.610 \times 10^0$	2.65	$3.449 \times 10^0$	2.63	$1.297 \times 10^1$	2.31

The error is computed as  $\|\Delta_{\parallel} K_M - (\Delta_{\parallel} K_M)_h\|$

objects, such as vesicles and biological cells [11]. Application of this sixth-order scheme to vesicle shape dynamics will be presented in the future.

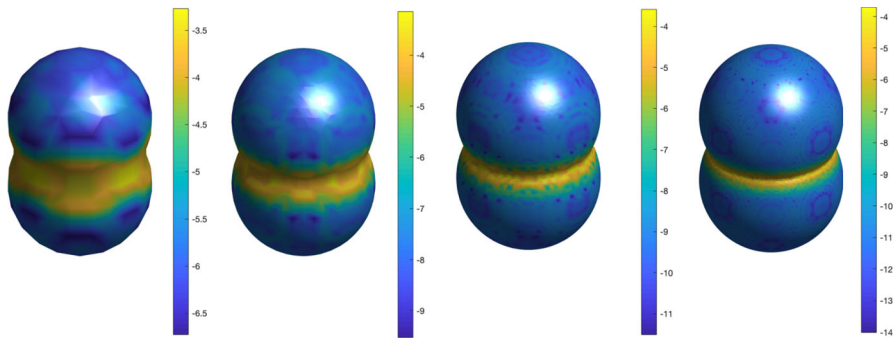
As a final test, we consider a surface with a kink. We initialize our level set function  $\phi(x, y, z)$  to be two merging spheres:

$$\phi(x, y, z) = \min((x^2 + y^2 + (z - z_0)^2 - r^2), (x^2 + y^2 + (z + z_0)^2 - r^2)), z_0 = 0.3, r = 0.6.$$

**Table 9** Accuracy results for the computation of surface Laplacian of the interface mean curvature for the red blood cell shape using fourth order accurate scheme from [7].

	$L_1$ -Error	Order	$L_2$ -Error	Order	$L_\infty$ -Error	Order
$16^3$	$2.672 \times 10^2$	–	$1.156 \times 10^2$	–	$9.802 \times 10^1$	–
$32^3$	$2.203 \times 10^2$	0.28	$1.096 \times 10^2$	0.08	$1.270 \times 10^2$	–0.37
$64^3$	$1.333 \times 10^2$	0.72	$7.319 \times 10^1$	0.58	$1.131 \times 10^2$	0.17
$128^3$	$1.338 \times 10^2$	–0.01	$7.785 \times 10^1$	–0.09	$1.296 \times 10^2$	–0.20

The error is computed as  $\|\Delta_{\parallel} K_M - (\Delta_{\parallel} K_M)_h\|$

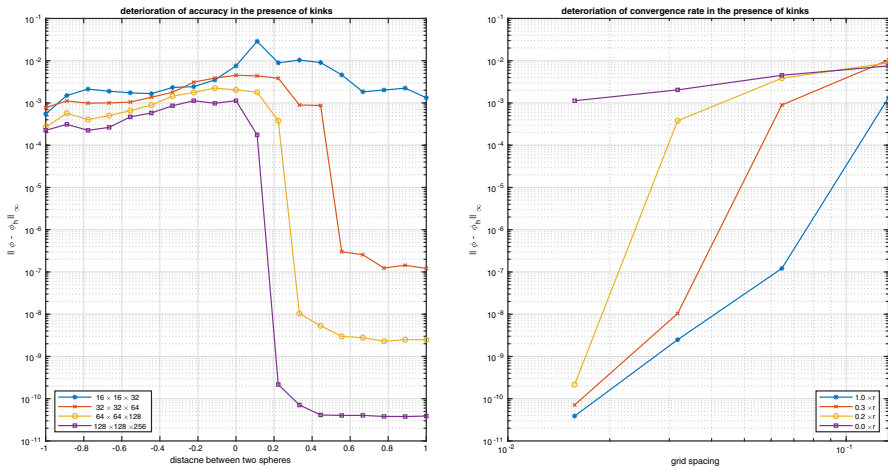


**Fig. 4** Distribution of numerical error  $\log_{10} |\phi - \phi_h|$  of the level set function on different grids. Grid size from left to right:  $16 \times 16 \times 32$ ,  $32 \times 32 \times 64$ ,  $64 \times 64 \times 128$ ,  $128 \times 128 \times 256$  (Color figure online)

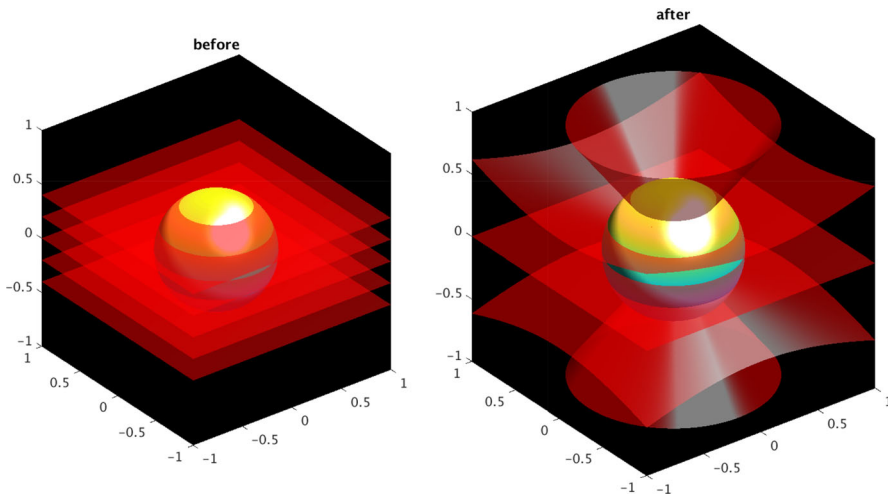
Then we reinitialize  $\phi(x, y, z)$  to be a signed distance map on different grids and plot  $\log_{10} |\phi_{\text{exact}} - \phi_{\text{numerical}}|$  on the surface, where  $\phi_{\text{exact}}$  is the exact signed distance map and  $\phi_{\text{numerical}}$  is the reinitialized  $\phi$ . As is shown in Fig. 4, near the kink, accuracy is greatly affected, but the rate of convergence is not affected away from the kink. Moreover, as finer grids are used, the affected region shrinks, as the inaccuracy arises due to not having a sufficient number of nodes to properly resolve the distance map with our interpolation scheme. As the grid spacing decreases, it is possible to resolve a larger fraction of the shape away from the kink. Figure 5 demonstrates the deterioration of accuracy for two approaching spheres. In the left graph of Fig. 5, the maximum error  $\|\phi - \phi_h\|_\infty$  is plotted as a function of the distance  $d = 2(z_0 - r)$  between the two spheres on different grids. A visible jump in error can be seen as the two spheres approach each other. The right graph of Fig. 5 shows that the order of accuracy for  $\|\phi - \phi_h\|_\infty$  decreases to first order, which is expected in the presence of kinks [20]. Adaptive mesh refinement or finite element methods [17] might help in the presence of kinks, but this is beyond the scope of this paper.

### 4.2 Extrapolation of Surface Scalar Fields in 3D

Extrapolation of fields living on a surface to the embedding space, i.e. solving Eq. (3), is of great importance in many applications. In the level set method, it is common practice to extend velocity fields defined only on the surface away from the interface in the normal direction. When solving PDEs on implicit surfaces represented by level set functions [1,31], dynamical fields have to be extrapolated to embed those PDEs in space. When those embedded PDEs



**Fig. 5** Left graph:  $\|\phi - \phi_h\|_\infty$  and distance  $d = 2z_0 - 2r$  for different grids. Convergence of  $\|\phi - \phi_h\|_\infty$  for different distances  $d = r, 0.3r, 0.2r, 0$  (Color figure online)



**Fig. 6** The  $(0, \pm 0.2, \pm 0.4)$  isosurface of  $\psi$  before and after extrapolation (Color figure online)

are of high orders, it becomes crucial to accurately extrapolate those surface fields [10]. However, this extrapolation is seldom done in an accurate way because boundary conditions on the implicit interface are seldom treated appropriately. Often the sign function is smoothed out in some fashion [10,31], but this regularization is not enough to prevent information from flowing across the interface, affecting the accuracy of the scheme globally. Note that in [10], the author shows that by using  $\text{Sign}(\phi) = \phi / \sqrt{\phi^2 + h^{1/2}}$  and the WENO scheme relatively small errors are obtained on a  $400^2$  grid. However, they did not show the convergence of their extrapolation scheme.

Here we test our ability to accurately extend fields away from a given surface. We consider a surface represented by the signed distance function  $\phi(x, y, z) = \sqrt{x^2 + y^2 + z^2} - 0.5$  on a  $[-1, 1]^3$  domain. Let a surface field  $\psi$  be the  $z$  coordinate of the position vector. We initialize

**Table 10** Accuracy results for the extended surface field

	$\ \psi - \psi_h\ _1$	Order	$\ \psi - \psi_h\ _2$	Order	$\ \psi - \psi_h\ _\infty$	Order
$16^3$	$1.794 \times 10^{-4}$	–	$1.138 \times 10^{-4}$	–	$1.324 \times 10^{-4}$	–
$32^3$	$4.843 \times 10^{-6}$	5.21	$4.014 \times 10^{-6}$	4.82	$6.734 \times 10^{-6}$	4.30
$64^3$	$3.517 \times 10^{-8}$	7.11	$3.341 \times 10^{-8}$	6.91	$2.074 \times 10^{-7}$	5.02
$128^3$	$5.611 \times 10^{-10}$	5.97	$5.345 \times 10^{-10}$	5.97	$2.932 \times 10^{-9}$	6.14

**Table 11** Accuracy results for the Laplacian of the extended surface field

	$\ \Delta\psi - (\Delta\psi)_h\ _1$	Order	$\ \Delta\psi - (\Delta\psi)_h\ _2$	Order	$\ \Delta\psi - (\Delta\psi)_h\ _\infty$	Order
$16^3$	$2.482 \times 10^{-2}$	–	$1.605 \times 10^{-1}$	–	$1.539 \times 10^{-1}$	–
$32^3$	$1.423 \times 10^{-3}$	4.12	$1.152 \times 10^{-2}$	3.80	$2.231 \times 10^{-2}$	2.73
$64^3$	$8.935 \times 10^{-4}$	3.99	$7.101 \times 10^{-4}$	4.02	$1.917 \times 10^{-3}$	3.60
$128^3$	$5.617 \times 10^{-5}$	3.99	$4.644 \times 10^{-5}$	3.93	$1.245 \times 10^{-4}$	3.94

**Table 12** Accuracy results for the BiLaplacian of the extended surface field

	$\ \Delta^2\psi - (\Delta^2\psi)_h\ _1$	Order	$\ \Delta^2\psi - (\Delta^2\psi)_h\ _2$	Order	$\ \Delta^2\psi - (\Delta^2\psi)_h\ _\infty$	Order
$16^3$	$3.511 \times 10^1$	–	$2.369 \times 10^1$	–	$2.675 \times 10^1$	–
$32^3$	$1.320 \times 10^1$	1.41	$1.035 \times 10^1$	1.19	$2.236 \times 10^1$	0.26
$64^3$	$1.810 \times 10^0$	2.87	$1.612 \times 10^0$	2.68	$4.035 \times 10^0$	2.47
$128^3$	$2.985 \times 10^{-1}$	2.60	$3.226 \times 10^{-1}$	2.32	$1.288 \times 10^0$	1.65

**Table 13** Accuracy results for the BiLaplacian of the extended surface field if ENO scheme from [7] is used to treat boundary terms

	$\ \Delta^2\psi - (\Delta^2\psi)_h\ _1$	Order	$\ \Delta^2\psi - (\Delta^2\psi)_h\ _2$	Order	$\ \Delta^2\psi - (\Delta^2\psi)_h\ _\infty$	Order
$16^3$	$3.627 \times 10^1$	–	$2.390 \times 10^1$	–	$2.765 \times 10^1$	–
$32^3$	$1.694 \times 10^1$	1.10	$1.671 \times 10^1$	0.52	$4.017 \times 10^1$	–0.54
$64^3$	$1.744 \times 10^1$	–0.04	$1.634 \times 10^1$	0.03	$4.005 \times 10^1$	0.00
$128^3$	$1.449 \times 10^1$	0.27	$1.675 \times 10^1$	–0.04	$6.863 \times 10^1$	–0.78

$\psi$  to be  $z$  at all points in space, and then solve the extension velocity equation [Eq. (15) with the Hamiltonian defined in Eq. (17)]. Figure 6 shows isosurfaces of  $\psi$  before and after extrapolation. The extrapolated surface field is determined to sixth-order accuracy (Table 10). Likewise, the Laplacian of  $\psi$  is fourth-order accurate (Table 11) and the BiLaplacian of the  $\psi$  is accurate to second-order (Table 12). If, instead, the ENO scheme from [7] is used, the computation of  $\Delta^2\psi$  does not converge (Table 13), which clearly shows the necessity of using a high order scheme to extrapolate any surface field whose dynamics depends on differential operators such as the Laplacian and the BiLaplacian. Indeed, the scheme presented here will be useful for a wide range of applications involving PDEs on curved surfaces.



### 4.3 Computation of Geodesic Distances on an Implicit Surface

Computation of geodesics is of interest in many applications [3,6]. A geodesic, though, is just a distance map defined on a surface. Solving for geodesics can therefore be regarded as a generalization of the reinitialization process to a non-Euclidean space. For example, consider a curve represented by the intersection of two level set functions  $\phi$  and  $\psi$ , where the zero level set of  $\phi$  represents the surface confining motion of the curve [3]. To compute the signed distance function on the surface for this curve, we need to iterate to equilibrium the following PDE on the surface [3]

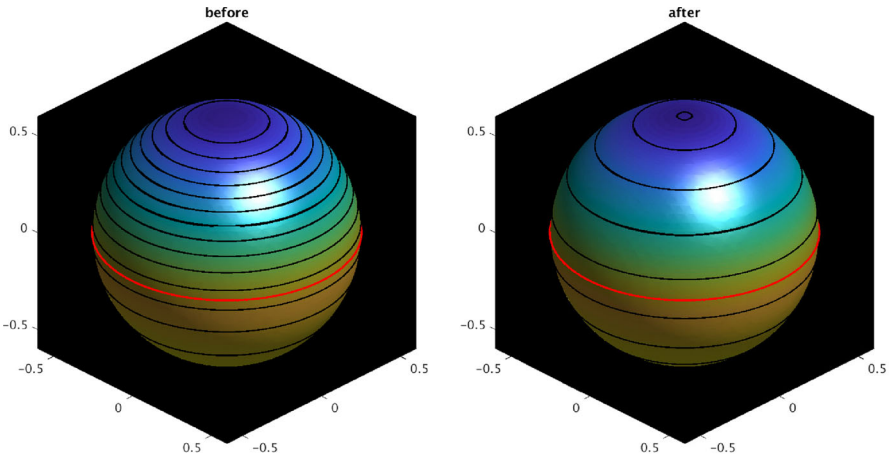
$$\frac{\partial \psi}{\partial \tau} + \text{Sign}(\psi^0) (|\nabla_{\parallel} \psi| - 1) = 0 \quad (23)$$

where  $\nabla_{\parallel}$  is the differential operator defined on the surface. Equation (23) is still a Hamilton–Jacobi equation. Due to the surface differential operator it is no longer feasible to use Godunov’s scheme to construct the numerical Hamiltonian. In [3], Local Lax–Friedrichs (LLF) schemes were used instead, and the author claims to find first-order accuracy in one dimension due to motion of the curve position during the iteration process. There are two sources of numerical error. First, the LLF scheme is more dissipative than Godunov’s scheme and will perturb the curve position. Second, the treatment of the boundary condition is not appropriately implemented, and information flows across curve. There is no easy way to remedy these problems, as far as we know.

Another way to solve Eq. (23) is to replace the surface differential operator  $\nabla_{\parallel}$  by the 3D Cartesian differential operator  $\nabla$ , using a closest point representation of  $\psi$  [19]. The closest point representation can be obtained by constraining the level sets of the distance map that defines the curve,  $\psi$ , to be perpendicular to the zero level set of  $\phi$ . That is, we want to solve the 3D reinitialization equation for  $\psi$ , subject to the condition that  $\nabla \phi \cdot \nabla \psi = 0$ , which is the same condition as for extending a field away from the surface. Therefore, we can simultaneously solve the extension equation (Eq. (3)) and the reinitialization equation (Eq. (2)) in order to determine the geodesic distance map  $\psi$ . We do this by iterating the extension equation (Eq. (3)) before each stage of the Runge–Kutta scheme in the reinitialization time stepping. In this way, we can use the standard reinitialization equation (Eq. (2)) in place of the more complicated non-Euclidean reinitialization equation (Eq. (23)).

Consider a surface represented by the signed distance function  $\phi(x, y, z) = \sqrt{x^2 + y^2 + z^2} - 0.6$  on a  $[-1, 1]^3$  domain. Let a closed curve  $\Gamma$  on this surface be the intersection of the zero level set of  $\phi$  and the zero of the function  $\psi(x, y, z) = e^{2z} - 1$ . We then reinitialize  $\psi$  using the procedure just described. Figure 7 shows the level curves of  $\psi$  before and after this redistancing. As shown in Table 14, our method provides sixth order accuracy for the computation of the geodesic distance between  $\Gamma$  and other points on the surface. In addition, we are able to calculate the geodesic curvature  $k_g$  of the level curves of  $\psi$  with fourth order accuracy (Table 15). Note that the order of accuracy for  $\psi$  seems to degenerate to fifth order as grids are refined (Table 14). This decrease in accuracy might result from accumulation of numerical errors from the extrapolation of  $\psi$  during each reinitialization step. This will be a subject of future research.

As a more stringent test, consider another example where the surface is still given by the signed distance function  $\phi(x, y, z) = \sqrt{x^2 + y^2 + z^2} - 0.6$  on a  $[-1, 1]^3$  domain, but the curve  $\Gamma$  is the intersection of the zero level set of  $\phi$  and the zero level set of  $\psi(x, y, z) = \exp \left[ 1.2 \left( \arcsin \left( \frac{z}{\sqrt{x^2 + y^2 + z^2}} \right) + \frac{\pi}{12} \right) \right]$ . Figure 8 shows the level curves of  $\psi$  before and after redistancing. Tables 16 and 17 show the results of our convergence tests for the geodesics



**Fig. 7** Level curves of  $\psi$  before and after redistancing. The red curve is the zero level curve of  $\psi$ . The spacing between neighboring black curves is  $\frac{20}{99}$  (Color figure online)

**Table 14** Accuracy results for the signed distance function on a curved surface

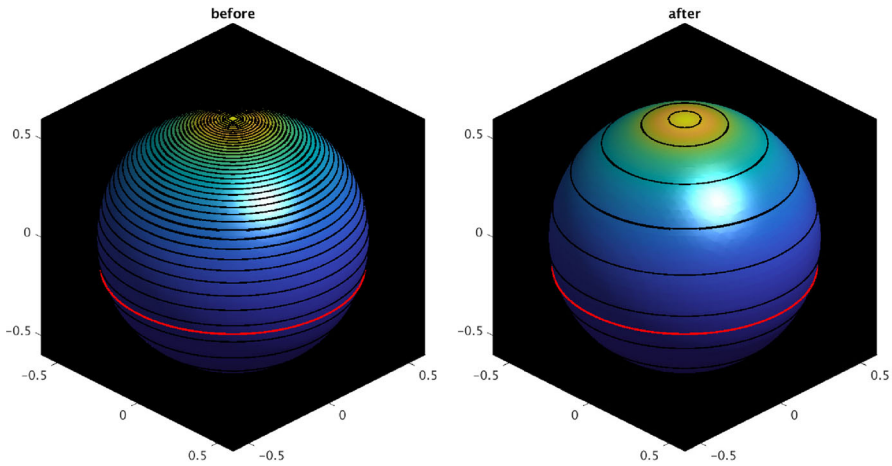
	$\ \psi - \psi_h\ _1$	Order	$\ \psi - \psi_h\ _2$	Order	$\ \psi - \psi_h\ _\infty$	Order
$16^3$	$3.166 \times 10^{-3}$	–	$2.221 \times 10^{-3}$	–	$3.320 \times 10^{-3}$	–
$32^3$	$4.246 \times 10^{-5}$	6.02	$3.327 \times 10^{-5}$	6.06	$5.229 \times 10^{-5}$	5.99
$64^3$	$1.310 \times 10^{-6}$	5.02	$9.151 \times 10^{-7}$	5.18	$1.448 \times 10^{-6}$	5.17
$128^3$	$4.312 \times 10^{-8}$	4.93	$2.928 \times 10^{-8}$	4.97	$6.897 \times 10^{-8}$	4.39

**Table 15** Accuracy results for the geodesic curvature for the level sets of  $\psi$

	$\ k_g - (k_g)_h\ _1$	Order	$\ k_g - (k_g)_h\ _2$	Order	$\ k_g - (k_g)_h\ _\infty$	Order
$16^3$	$2.772 \times 10^{-1}$	–	$1.851 \times 10^{-1}$	–	$2.147 \times 10^{-1}$	–
$32^3$	$2.723 \times 10^{-2}$	3.35	$1.987 \times 10^{-2}$	3.22	$2.655 \times 10^{-2}$	3.02
$64^3$	$8.437 \times 10^{-4}$	5.01	$9.341 \times 10^{-4}$	4.41	$3.100 \times 10^{-3}$	3.10
$128^3$	$3.035 \times 10^{-5}$	4.80	$4.190 \times 10^{-5}$	4.48	$3.867 \times 10^{-4}$	3.00

and the geodesic curvature, respectively. It seems that the order of convergence degrades as the grid is refined. The convergence rate, however, is still better than that in [3]. This loss of convergence rate is beyond the scope of current work and requires further investigation of our implementation of the closest point method [19]. Still, this scheme can be useful in applications dealing with the motion of curves embedded in a surface driven by geodesic curvatures [3].

As a final test, we compute geodesic distances on the Stanford bunny [29] (Fig. 9). The level set representation of the bunny is obtained by the radial function method [8] on a  $96^3$  grid and is then smoothed by diffusion. The function  $\psi(x, y, z) = \exp(x^2 + y^2 + (z - 0.05)^2) - \exp(0.0009)$  is used to initialize curve position on the bunny. This example shows that we can compute geodesic distances on highly curved surfaces.



**Fig. 8** Level curves of  $\psi$  before and after redistancing. The red curve is the zero level curve of  $\psi$ . The spacing between neighboring black curves is  $\frac{20}{99}$  (Color figure online)

**Table 16** Accuracy results for the signed distance function on a curved surface

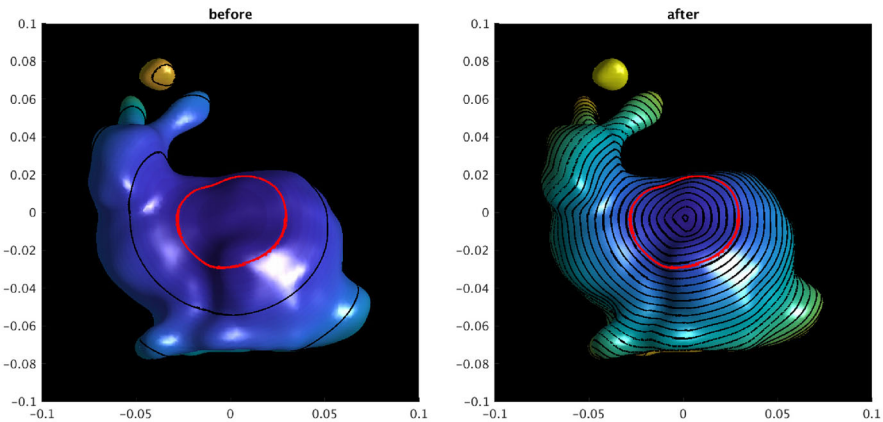
	$\ \psi - \psi_h\ _1$	Order	$\ \psi - \psi_h\ _2$	Order	$\ \psi - \psi_h\ _\infty$	Order
$16^3$	$9.154 \times 10^{-4}$	–	$7.693 \times 10^{-4}$	–	$2.620 \times 10^{-3}$	–
$32^3$	$8.714 \times 10^{-5}$	3.39	$7.777 \times 10^{-5}$	3.31	$1.400 \times 10^{-4}$	4.23
$64^3$	$8.170 \times 10^{-6}$	3.41	$7.002 \times 10^{-6}$	3.47	$1.415 \times 10^{-5}$	3.31
$128^3$	$2.549 \times 10^{-6}$	1.68	$2.226 \times 10^{-6}$	1.65	$4.114 \times 10^{-6}$	1.78

**Table 17** Accuracy results for the geodesic curvature for the level sets of  $\psi$

	$\ k_g - (k_g)_h\ _1$	Order	$\ k_g - (k_g)_h\ _2$	Order	$\ k_g - (k_g)_h\ _\infty$	Order
$16^3$	$1.337 \times 10^{-1}$	–	$1.233 \times 10^{-1}$	–	$1.885 \times 10^{-1}$	–
$32^3$	$2.500 \times 10^{-2}$	2.42	$2.089 \times 10^{-2}$	2.56	$3.548 \times 10^{-2}$	2.41
$64^3$	$1.217 \times 10^{-2}$	1.04	$1.195 \times 10^{-2}$	0.81	$2.617 \times 10^{-2}$	0.44
$128^3$	$8.174 \times 10^{-3}$	0.57	$7.740 \times 10^{-3}$	0.64	$2.245 \times 10^{-2}$	0.22

## 5 Conclusion

In many applications of science and engineering, it is important to accurately compute surface curvature and its derivatives in three dimensional space. In problems involving elastic surfaces, second derivatives of surface curvatures such as  $\Delta_{\parallel} K_M$  are needed. Many widely used schemes are unable to compute this geometric quantity accurately. For instance, all schemes reviewed in [11] are based on surface triangulation and fail to compute  $\Delta_{\parallel} K_M$  convergently. Other level set based schemes such as those proposed in [5] are difficult to generalize to three dimensional cases. We approached this problem in the level set framework by maintaining accuracy of the level set function in the reinitialization process.



**Fig. 9** Geodesic distances computed on the Stanford bunny. The left panel shows the equipotential intersections of  $\psi$  before redistancing, and the right panel shows the geodesics computed by our redistancing procedure. The red curve is the intersection of the zero level set of  $\psi$  with the bunny whose position is preserved during the redistancing procedure. The spacing between neighborhood black curves is  $\frac{1}{245}$  (Color figure online)

In particular, we have presented a sixth-order accurate numerical scheme for Hamilton–Jacobi equations with a level set-defined boundary condition. This work builds on the work of Ch  n   and Min [7]. We showed that our method can solve the reinitialization equation of Sussman [28] and the extrapolation equation (3) to sixth order accuracy in the  $L_1$ ,  $L_2$ , and  $L_\infty$  norms for smooth surfaces. Our numerical experiments also show that this method leads to an interface curvature that is accurate to fourth order, which results in second order accuracy of bending forces calculated for elastic surfaces. These schemes thus make possible an accurate simulation of dynamical elastic surfaces in the level set framework and of other applications in physics and engineering that require accurate computation of interface curvature. Also, a sixth order accurate extrapolation scheme for surface fields is proposed, which allows a very accurate closest point representation [19] of surface fields defined near the interface, which can be a crucial component for embedding high order PDEs on a non-Euclidean surface in space [10] and in solving the level set equation [32]. Finally, we presented a convergent method for computing geodesics on an implicit surface. The combination of techniques presented here can greatly improve the accuracy of simulations involving elastic surfaces and also makes possible the simulation of the dynamics of biomembranes coupled to concentrations of interacting, surface-bound proteins, as occurs in processes such as endo- and exocytosis, cell division, and cell motility.

## References

1. Adalsteinsson, D., Sethian, J.A.: Transport and diffusion of material quantities on propagating interfaces via level set methods. *J. Comput. Phys.* **185**(1), 271–288 (2003)
2. Bardi, M., Osher, S.: The nonconvex multidimensional Riemann problem for Hamilton–Jacobi equations. *SIAM J. Math. Anal.* **22**(2), 344–351 (1991)
3. Cheng, L.-T., Burchard, P., Merriman, B., Osher, S.: Motion of curves constrained on surfaces using a level-set approach. *J. Comput. Phys.* **175**(2), 604–644 (2002)
4. Chopp, D., Sethian, J.A.: Motion by intrinsic Laplacian of curvature. *Interfaces Free Bound.* **1**(1), 107–123 (1999)
5. Coquerelle, M., Glockner, S.: A fourth-order accurate curvature computation in a level set framework for two-phase flows subjected to surface tension forces. *J. Comput. Phys.* **305**, 838–876 (2016)

6. Crane, K., Weischedel, C., Wardetzky, M.: Geodesics in heat: a new approach to computing distance based on heat flow. *ACM Trans. Graph. (TOG)* **32**(5), 152 (2013)
7. du Ch n , A., Min, C., Gibou, F.: Second-order accurate computation of curvatures in a level set framework using novel high-order reinitialization schemes. *J. Sci. Comput.* **35**(2–3), 114–131 (2008)
8. Fasshauer, G.E.: *Meshfree Approximation Methods with MATLAB*, vol. 6. World Scientific, Singapore (2007)
9. Goldman, R.: Curvature formulas for implicit curves and surfaces. *Comput. Aided Geom. Des.* **22**(7), 632–658 (2005)
10. Greer, J.B., Bertozzi, A.L., Sapiro, G.: Fourth order partial differential equations on general geometries. *J. Comput. Phys.* **216**(1), 216–246 (2006)
11. Guckenberger, A., Schraml, M.P., Chen, P.G., Leonetti, M., Gekle, S.: On the bending algorithms for soft objects in flows. *Comput. Phys. Commun.* **207**, 1–23 (2016)
12. Harten, A., Osher, S.: Uniformly high-order accurate nonoscillatory schemes. I. *SIAM J. Numer. Anal.* **24**(2), 279–309 (1987)
13. Helfrich, W.: Elastic properties of lipid bilayers: theory and possible experiments. *Zeitschrift f r Naturforschung C* **28**(11–12), 693–703 (1973)
14. Jiang, G.-S., Peng, D.: Weighted ENO schemes for Hamilton–Jacobi equations. *SIAM J. Sci. Comput.* **21**(6), 2126–2143 (2000)
15. Jiang, G.-S., Shu, C.-W.: Efficient implementation of weighted ENO schemes. *J. Comput. Phys.* **126**(1), 202–228 (1996)
16. Laadhari, A., Saramito, P., Misbah, C., Sz kely, G.: Fully implicit methodology for the dynamics of biomembranes and capillary interfaces by combining the level set and Newton methods. *J. Comput. Phys.* **343**, 271–299 (2017)
17. Lipnikov, K., Morgan, N.: A high-order discontinuous Galerkin method for level set problems on polygonal meshes. *J. Comput. Phys.* **397**, 108834 (2019)
18. Liu, X.-D., Osher, S., Chan, T., et al.: Weighted essentially non-oscillatory schemes. *J. Comput. Phys.* **115**(1), 200–212 (1994)
19. Macdonald, C.B., Ruuth, S.J.: Level set equations on surfaces via the closest point method. *J. Sci. Comput.* **35**(2–3), 219–240 (2008)
20. Min, C.: On reinitializing level set functions. *J. Comput. Phys.* **229**(8), 2764–2772 (2010)
21. Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations. *J. Comput. Phys.* **79**(1), 12–49 (1988)
22. Peng, D., Merriman, B., Osher, S., Zhao, H., Kang, M.: A PDE-based fast local level set method. *J. Comput. Phys.* **155**(2), 410–438 (1999)
23. Russo, G., Smereka, P.: A remark on computing distance functions. *J. Comput. Phys.* **163**(1), 51–67 (2000)
24. Salac, D., Miksis, M.: A level set projection model of lipid vesicles in general flows. *J. Comput. Phys.* **230**(22), 8192–8215 (2011)
25. Sezgin, E., Levental, I., Mayor, S., Eggeling, C.: The mystery of membrane organization: composition, regulation and roles of lipid rafts. *Nat. Rev. Cell Biol.* **18**(1), 361–374 (2017)
26. Shu, C.-W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.* **77**(2), 439–471 (1988)
27. Smit, J., van Sint Annaland, M., Kuipers, J.A.M.: Grid adaptation with weno schemes for non-uniform grids to solve convection-dominated partial differential equations. *Chem. Eng. Sci.* **60**(10), 2609–2619 (2005)
28. Sussman, M., Smereka, P., Osher, S.: A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.* **114**(1), 146–159 (1994)
29. The Stanford 3D Scanning Repository. <http://graphics.stanford.edu/data/3Dscanrep/>
30. Towers, J.D.: Two methods for discretizing a delta function supported on a level set. *J. Comput. Phys.* **220**(2), 915–931 (2007)
31. Xu, J.-J., Zhao, H.-K.: An Eulerian formulation for solving partial differential equations along a moving interface. *J. Sci. Comput.* **19**(1–3), 573–594 (2003)
32. Zhao, H.-K., Chan, T., Merriman, B., Osher, S.: A variational level set approach to multiphase motion. *J. Comput. Phys.* **127**(1), 179–195 (1996)
33. Zhong-can, O.-Y., Helfrich, W.: Bending energy of vesicle membranes: general expressions for the first, second, and third variation of the shape energy and applications to spheres and cylinders. *Phys. Rev. A* **39**, 5280–5288 (1989)