Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 210 (2005) 342–367
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A Lagrangian particle level set method

Simone E. Hieber, Petros Koumoutsakos *

Institute of Computational Science, ETH Zürich, CH-8092, Switzerland

Received 3 June 2004; received in revised form 15 April 2005; accepted 21 April 2005
Available online 16 June 2005

Abstract

We present a novel particle level set method for capturing interfaces. The level set equation is solved in a Lagrangian frame using particles that carry the level set information. A key aspect of the method involves a consistent remeshing procedure for the regularization of the particle locations when the particle map gets distorted by the advection field. The Lagrangian description of the level set method is inherently adaptive and exact in the case of solid body motions. The efficiency and accuracy of the method is demonstrated in several benchmark problems in two and three dimensions involving pure advection and curvature induced motion of the interface. The simplicity of the particle description is shown to be well suited for real time simulations of surfaces involving cutting and reconnection as in virtual surgery environments.

Keywords: Lagrangian particle methods; Level set method; Interface capturing

1. Introduction

The accurate and efficient simulation of interface evolution is of fundamental importance for a wide range of problems ranging from multiphase flows and combustion to virtual surgery environments and computer animation. In these applications we can distinguish two broad classes of computational methods used to describe the evolution of interfaces, namely: interface capturing and interface tracking methods.

In *capturing methods*, the interface is determined by an implicit function that is advected in the computational domain. The most common interface capturing methods include Volume of Fluid [1] and

* Corresponding author. Tel.: +411 632 5258; fax: +411 632 1703.
E-mail addresses: hieber@inf.ethz.ch (S.E. Hieber), petros@inf.ethz.ch (P. Koumoutsakos).
URL: http://www.icos.ethz.ch/cse/ (P. Koumoutsakos).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2005.04.013

Level Set methods [2,3]. Volume of Fluid (VOF) methods are inherently linked to fluid mechanics problems and to Eulerian discretizations of the flow equations. They have enjoyed significant success in simulations of free surface and multiphase flow phenomena [4]. Level Set (*LS*) methods [5–10] employ an implicit function to describe the advection of the interface and are well suited to problems where interfaces undergo extreme topological changes. They have been applied with significant success to problems ranging from fluid mechanics to image processing and materials science (see the textbooks [7,6] and references therein). The LS equation is commonly solved in an Eulerian framework by using high order finite difference methods, such as the fifth-order accurate Hamilton–Jacobi WENO schemes [11]. The accuracy of interface capturing schemes is reduced when the interface develops structures whose length scales are smaller than those afforded by the Eulerian mesh [12]. In addition time step limitations are introduced by the associated CFL condition for the discretization of the advection term. A number of remedies have been proposed to rectify this situation, such as high-order ENO/WENO approximations, semi-Lagrangian techniques [13] and hybrid particle-level set techniques as introduced in [14]. In the latter work, the cells near the interface are seeded with marker particles in order to obtain subgrid scale accuracy. This hybrid method has been shown to provide superior results for a number of benchmark problems in two and three dimensions. However, a number of open issues remain regarding the manner in which particles are introduced as well as the number of particles necessary to obtain a prescribed accuracy.

Tracking methods solve the interface evolution equation in a Lagrangian fashion, for example by evolving marker particles. The origin of tracking methods can be traced to the 1930s and to calculations made by hand by Rosenhead [15] to describe the evolution of a vortex sheet in incompressible flows. These calculations have been followed 40 years later by the introduction of vortex methods [16] and the method of contour dynamics [17]. A fundamental problem of Lagrangian methods is the distortion of the locations of the computational elements resulting in an inaccurate description of the interface. A regularization procedure is necessary in order to compensate for this defect and to maintain the accuracy of the method. In the context of LS methods rather ad-hoc procedures, such as particle insertion and deletion, have been proposed and in [14] it is argued that no a-priori way exists that may help in building a regularization of these methods.

In this paper we present a novel particle level set method that overcomes the difficulties associated with the Lagrangian formulation of level set equations, using techniques originally developed for vortex particle methods. In the particle framework the level set function is mollified using suitable kernel functions and it is subsequently discretized using particles as quadrature points. The level set information can be recovered by a linear superposition of the quantities carried by the individual particles. The accuracy of the method requires that particles overlap (i.e. their distance is smaller than their core size) at all times [18]. This is ensured by a remeshing procedure as it was first introduced in the context of vortex particle methods [19].

The proposed method enjoys the advantages of particle methods such as adaptivity and unconditional stability for the advection of the computational elements. The description of the level set function as a linear superposition of individual particle functions enables straightforward computations of interface normals and curvatures. This enables computations with curvature induced motion of the interface. In addition the simplicity of the method allows real time simulations involving the cutting and reconnection of interfaces.

The outline of the present paper is as follows: In Section 2, we describe the level set method followed by its particle representation in Section 3. The remeshing of the particles as well as the reinitialization of the level set function is discussed in Section 4. In Section 5, we present the validation of our method by considering two and three dimension benchmark problems and we discuss its use in virtual surgery environments. The paper concludes with an assessment of the method and outlines future work.

2. Level set method

The Level Set method [2,3] defines an interface $\Gamma(t)$ as the zero level set of a high dimensional, scalar function $\Phi(\mathbf{x}, t): \mathbb{R}^3 \to \mathbb{R}$:

$$\Gamma(t) = \{\mathbf{x} \in \Omega : \Phi(\mathbf{x}, t) = 0\}, \tag{1}$$

where Ω is the computational domain. The level set function has the following properties:

$$\begin{aligned} &\Phi(\mathbf{x}, t) > 0, \quad x \in \tilde{\Omega}, \\ &\Phi(\mathbf{x}, t) \leqslant 0, \quad x \notin \tilde{\Omega}, \end{aligned} \tag{2}$$

where $\tilde{\Omega} \subset \Omega$ is an open region bounded by Γ. The motion of the interface is driven by a velocity field $\mathbf{u}(\mathbf{x}, t)$ as

$$\frac{\partial \Phi}{\partial t} + \mathbf{u} \cdot \nabla \Phi = 0 \quad \text{for } t > 0, \tag{3}$$

$$\Phi(\mathbf{x}, 0) = \Phi_0(\mathbf{x}). \tag{4}$$

The specific form of the velocity field depends on the problem under consideration.

The function Φ_0 is usually chosen as the signed distance to the interface such that $|\nabla \Phi| = 1$. However, during its evolution, the level set function $\Phi(t)$ can lose the property of being the distance function [20]. Reinitialization schemes such as fast marching methods have been introduced [3,21] in order to maintain this property. Usually the evolution of the level set function is computed using grid-based methods and the spatial derivatives in Eqs. (7) and (8) are calculated by finite difference schemes [7].

Alternatively the level set equation can be expressed in a Lagrangian framework using the material derivative $\frac{\mathrm{D}}{\mathrm{D}t} = \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla$ as

$$\frac{\mathrm{D}\Phi}{\mathrm{D}t} = 0, \tag{5}$$

$$\frac{\mathrm{D}\mathbf{x}}{\mathrm{D}t} = \mathbf{u}, \tag{6}$$

where $\mathbf{x}$ denotes the characteristics of the equation. The Lagrangian description of the level set equation is utilized in interface tracking methods. These methods encounter difficulties when singularities are formed during the evolution of the interface and need to be complemented with suitable regularization procedures [3] in order to recover a desired weak solution. In this work this regularization is performed by a remeshing procedure (see Section 4.1).

The unit normal and the curvature of the interface can be evaluated based on the level set function by

$$\mathbf{n} = \frac{\nabla \Phi}{|\nabla \Phi|}, \tag{7}$$

$$\kappa = \nabla \cdot \frac{\nabla \Phi}{|\nabla \Phi|}. \tag{8}$$

The calculation of interface normals and curvature involves the computation of gradients of the level set function which is achieved on the mesh in grid based methods, whereas in the present particle methods they are calculated by taking suitable derivatives of the mollification kernels (see Section 3.1.1).

3. Particle representation of level sets

In this paper we implement two descriptions of level set functions, namely: the signed distance function and the color function.

The *signed distance function* (SDF) is defined by Eq. (2) along with the constraint that

$$|\nabla \Phi(\mathbf{x}, t)| = 1. \tag{9}$$

The absolute value of the SDF measures the distance to the interface and the sign of the function changes when crossing the interface.

The *color function* (CF) [22] is defined by a different characteristic constant on each subdomain separated by the interface. The CF used in this work is

$$\Phi(\mathbf{x}, t) = 1, \quad \mathbf{x} \in \tilde{\Omega}, \tag{10}$$

$$\Phi(\mathbf{x}, t) = 0, \quad \mathbf{x} \notin \tilde{\Omega}, \tag{11}$$

where $\tilde{\Omega} \subset \Omega$ is an open region bounded by the interface Γ.

In level set methods the SDF approach can be used for computing interface quantities such as surface tension. However, in cases where the distance information is not necessary, use of the CF can result in significant computational savings.

3.1. Level set function and gradient approximations using particles

In the context of particle methods [18,23,16] a smooth approximation of the level set function can be constructed by using a mollification kernel $\zeta_\epsilon(\mathbf{x})$:

$$\Phi_\epsilon(\mathbf{x}) = \Phi \bigstar \zeta_\epsilon = \int \Phi(\mathbf{y}) \zeta_\epsilon(\mathbf{x} - \mathbf{y}) \, \mathrm{d}\mathbf{y}, \tag{12}$$

where ϵ denotes a characteristic length of the kernel.

The kernel is said to be of order r when the following moment conditions [18] are satisfied:

$$\int \zeta_\epsilon(\mathbf{x}) \, \mathrm{d}\mathbf{x} = 1, \tag{13}$$

$$\int \mathbf{x}^i \zeta_\epsilon(\mathbf{x}) \, \mathrm{d}\mathbf{x} = 0 \quad \text{if } |i| \leqslant r - 1, \tag{14}$$

$$\int |\mathbf{x}|^r \zeta_\epsilon(\mathbf{x}) \, \mathrm{d}\mathbf{x} < \infty. \tag{15}$$

This mollified approximation $\Phi_\epsilon(\mathbf{x})$ can be discretized using the particle locations as quadrature points and a particle approximation of the regularized level set function is

$$\Phi^h_\epsilon(\mathbf{x}) = \Phi^h \bigstar f \zeta_\epsilon = \sum_{p=1}^{N} v_p \Phi_p \zeta_\epsilon(\mathbf{x} - \mathbf{x}_p), \tag{16}$$

where $\mathbf{x}_p$ and v_p denote the position and volume of the pth particle, and $\Phi_p = \Phi(\mathbf{x}_p)$ the value of the level set at the $p = 1, \ldots, N$ particle locations.

As discussed in [18] the error introduced by the quadrature of the mollified approximation of Φ can be distinguished in two parts as

$$\Phi - \Phi^h_\epsilon = (\Phi - \Phi \bigstar f \zeta_\epsilon) + (\Phi - \Phi^h) \bigstar f \zeta_\epsilon. \tag{17}$$

The first term in Eq. (17) denotes the mollification error that can be controlled by appropriately selecting the kernel properties. The second term denotes the quadrature error due to the approximation of the integral on the particle locations. The overall accuracy of the method [18] results in

$$\|\Phi - \Phi_\epsilon^h\|_{0,p} \leqslant \|\Phi - \Phi_\epsilon\|_{0,p} + \|\Phi_\epsilon - \Phi_\epsilon^h\|_{0,p} \sim \mathcal{O}(\epsilon^r) + \mathcal{O}\left(\frac{h^m}{\epsilon^m}\right), \tag{18}$$

where $\|(\cdot)\|_{0,p} = (\int(\cdot)^p \,\mathrm{d}\mathbf{x})^{1/p}$ and r denotes the order of the first non-vanishing moment of the kernel ζ_ϵ [18]. For equidistant particle locations $m = \infty$ and for positive kernels such as the Gaussian, $r = 2$. Here for ζ_ϵ a quartic spline kernel with second order of accuracy is implemented:

$$\zeta_\epsilon(\mathbf{x}) = n_d \bar{\zeta}_\epsilon = n_d \begin{cases} \frac{s^4}{4} - \frac{5s^2}{8} + \frac{115}{192}, & 0 \leqslant s < \frac{1}{2}, \quad s = \frac{|\mathbf{x}|}{\epsilon}, \\ -\frac{s^4}{6} + \frac{5s^3}{6} - \frac{5s^2}{4} + \frac{5s}{24} + \frac{55}{96}, & \frac{1}{2} \leqslant s < \frac{3}{2}, \\ \frac{(2.5-s)^4}{24}, & \frac{3}{2} \leqslant s < \frac{5}{2}, \\ 0, & s \geqslant \frac{5}{2}. \end{cases} \tag{19}$$

The normalization value n_d depends on the dimension of the problem and is computed as

$$n_d = \frac{1}{\sum_j v_j \bar{\zeta}_\epsilon(\mathbf{x} - \mathbf{x}_j)} \tag{20}$$

ensuring the property of *partition of unity* for the particles. Kernels of arbitrary order [24] are possible by giving up the positivity of the kernel function.

The error estimates reveal a very important fact for smooth particle approximations. In order to obtain accurate approximations of the level set *smooth particles must overlap*. Note that the moment conditions expressed by the integrals of the mollifier functions are not often well represented in the case of discrete particle sets. These moment conditions can be ensured by appropriate normalizations [18].

3.1.1. Particle derivative approximations

Particle approximations of the derivative operators can be constructed through their integral approximations. This can be easily achieved by taking the derivatives of Eq. (12) as convolution and derivative operators commute in unbounded or periodic domains. This approximation is popular in particle methods such as Smooth Particle Hydrodynamics (SPH) [25] where derivatives of a field quantity Φ on a particle q are approximated in a conservative form as

$$\langle \nabla \Phi \rangle_q = \sum_p v_p (\Phi_p - \Phi_q) \nabla \zeta_\epsilon(\mathbf{x}_q - \mathbf{x}_p), \tag{21}$$

$$\langle \nabla^2 \Phi \rangle_q = \sum_p v_p (\Phi_p - \Phi_q) \nabla^2 \zeta_\epsilon(\mathbf{x}_q - \mathbf{x}_p). \tag{22}$$

The normalization values $n_{d,1}, n_{d,2}$ of $\nabla\zeta_\epsilon(\mathbf{x}) = n_{d,1}\nabla\bar{\zeta}_\epsilon(\mathbf{x})$ and $\nabla^2\zeta_\epsilon(\mathbf{x}) = n_{d,2}\nabla^2\bar{\zeta}_\epsilon(\mathbf{x})$ are chosen such that the corresponding non-zero moment condition [26] is satisfied. The kernel of Eq. (19) has its first three derivatives continuous allowing a smooth approximation of the spatial derivatives of $\Phi(\mathbf{x})$. The computation of interface curvature relies in employing these formulas for the computation of derivatives as defined in Eq. (8). An alternative formulation involves the development of integral operators that are equivalent to differential operators [26] as they were first introduced for the integral approximation of the Laplacian [27] in the diffusion equation.

4. Lagrangian particle level set

In the proposed Lagrangian method the evolution of the LS function Φ amounts to evolving the particles on which it is discretized. The particle position $\mathbf{x}_p$, volume v_p and level set attribute Φ_p, evolve by the following system of ordinary differential equations derived from Eq. (3):

$$\begin{aligned}\frac{\mathrm{D}\Phi_p}{\mathrm{D}t} &= 0,\\ \frac{\mathrm{D}v_p}{\mathrm{D}t} &= \langle \nabla \cdot \mathbf{u} \rangle_p v_p,\\ \frac{\mathrm{D}\mathbf{x}_p}{\mathrm{D}t} &= \mathbf{u}_p,\end{aligned} \tag{23}$$

where $\langle \diamond \rangle_p$ denotes the derivative approximation on a particle p (cf. Eq. (21)).

An immediate implication of the Lagrangian description is that simulation of solid body rotation is nearly exact, except for the introduction of errors introduced by the particle initialization and by the accuracy of the time integration. The spatial derivatives used in Eqs. (7) and (8) are computed according to the weighted sum over all particles of Eqs. (21) and (22).

4.1. Remeshing

A key aspect of the present method involves the use of a remeshing procedure. In smooth particle methods, as discussed earlier, particles must overlap at all times in order to guarantee the convergence of the method [28]. As it is shown in [29] remeshing is equivalent to a regularization of the particle description of the advected quantities.

In this work remeshing is employed in order to regularize the distorted particle locations and to redistribute accordingly particle quantities onto a uniform set of particles with the spacing h. The redistribution of particle quantities is achieved using the third-order M'_4 kernel [30] which in one dimension it is expressed as

$$M'_4(x,h) = \begin{cases} 1 - \frac{5s^2}{2} + \frac{3s^3}{2}, & 0 \leqslant s < 1, \quad s = \frac{|x|}{h},\\ \frac{(1-s)(2-s)^2}{2}, & 1 \leqslant s < 2,\\ 0, & s \geqslant 2. \end{cases} \tag{24}$$

In higher dimensions the interpolation formulas are tensorial products of their one-dimensional counterparts.

Remeshing is necessary only when particles cease to overlap as they adapt to the flow map. In order to determine the rate at which particle remeshing is necessary we introduce a measure of distortion. This measure relies on the fact that the weighted sum $H(t)$ over all particles must be equal to unity in a regularized particle map

$$H(t) = \sum_j v_j(t)\zeta_\epsilon(\mathbf{x}_p(t) - \mathbf{x}_j(t)), \tag{25}$$

$$H(0) = H_0 = 1. \tag{26}$$

The average change of $H(t)$ over all particles is a measurement of distortion

$$\Delta H = \frac{1}{N_p}\sum_j \frac{|H_j(t) - H_{0,j}|}{H_{0,j}}, \tag{27}$$

where N_p is the number of particles. When considering a purely rigid body motion, e.g. the rotation of Zalesak's disk, ΔH is zero. In our simulations remeshing is invoked each time the function ΔH exceeds a small prespecified threshold.

Remeshing acts to suppress the evolution of scales that are smaller than the particle core and to prevent the formation of spurious scales resulting from non-overlapping particles, thus providing an entropy condition for the evolution of the level sets.

4.2. Reinitialization for particle level sets

During its evolution, the level set function usually ceases to be the signed distance function. Techniques such as fast marching methods [31,5,3] and re-distancing algorithms [32] have been introduced in order to maintain this property by reinitializing the level set function.

A prerequisite for applying techniques such as the fast marching method is the regularity of the computational elements. This reinitialization is straightforward when using an Eulerian description of the level set methods but , in general, it is not possible for an arbitrary particle distribution. Remeshing, however, offers the benefit that it distributes the particles on a Cartesian mesh and it allows the implementation of the fast marching method.

Remeshing also enables use of the level set redistancing algorithm introduced by Sussman [32] by solving the following equation on the regularized particle locations:

$$\Phi_t = \mathrm{sign}(\Phi_0)(1 - |\nabla \Phi|), \Phi(\mathbf{x}, 0) = \Phi_0(\mathbf{x}) \quad \text{for } t \to \infty. \tag{28}$$

The computational effort of this scheme can be significantly high when the time integration of Eq. (28) requires a small time step to ensure the convergence of the solution.

An alternative scheme, well suited for particle methods as it does not require that particles are distributed on a regular mesh, was developed based on the first-order approximation of the derivative $\frac{\partial \Phi(\mathbf{x},t)}{\partial \mathbf{x}} = \frac{\Phi(\mathbf{x},t)-\Phi(\mathbf{x}_0,t)}{\mathbf{x}-\mathbf{x}_0}$, where $\mathbf{x}_0$ is the position on the interface that minimizes $|\mathbf{x} - \mathbf{x}_0|$. Reformulation of this equations leads to a first-order approximation of the distance to the interface that can be used in turn for reinitialization:

$$|\mathbf{x} - \mathbf{x}_0| = \left| \frac{\Phi(\mathbf{x}, t) - \Phi(\mathbf{x}_0, t)}{\frac{\partial \Phi(\mathbf{x},t)}{\partial \mathbf{x}}} \right| = \left| \frac{\Phi(\mathbf{x}, t)}{\frac{\partial \Phi(\mathbf{x},t)}{\partial \mathbf{x}}} \right|. \tag{29}$$

The approximation of the gradient of the level set that can be obtained on the particle locations using Eq. (21). We wish to mention here that the scheme was first proposed independently by Cottet [33] while we have also been recently aware of a similar work in [34].

We compare these schemes on the evolution of a one-dimensional signed distance function. This function evolves with a non-uniform velocity field leading to loss of its signed distance property (Fig. 1). The Taylor series re-distancing of Eq. (29) is computationally efficient and it maintains the signed distance property very close to the interface (within two particle spacing) but fails in regions further away from the interface as high-order derivatives become important. The redistancing scheme of [32], provides a good reconstruction within five iterations. The fast marching method yields the best results for the longer part of the domain and is overall the most efficient method in this test case. In the remaining of this paper we use the fast marching method for the re-initialization.

Our implementation of the fast marching method is based on the work of Sethian [21] and Adalsteinsson and Sethian [35]. In [35] the authors, illustrate how to find the initial set of level set values close to zero level set (Close points) to start the algorithm with. We extend their technique [35] in the case (a) where only one of the neighboring points of the considered close point is on the other side of the zero level set (Fig. 2). In this case, we do not define the level set value as the distance to the intersection point on the line connecting

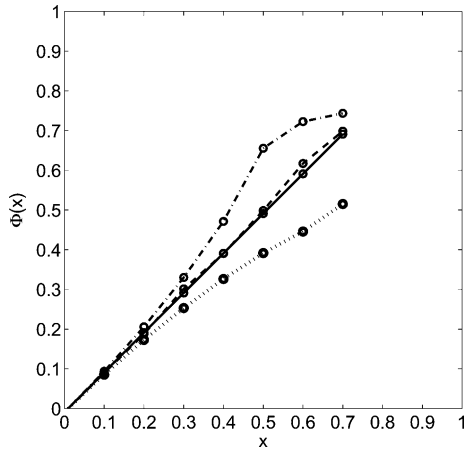

Fig. 1. Comparison of the reinitialization methods in one dimension. A distorted signed-distance function (dotted line) is reinitialized by a Taylor series redistancing algorithm (Eq. (29)) (dash–dot line), the redistancing algorithm [32] (dashed line) and the fast marching method [7] (solid line).

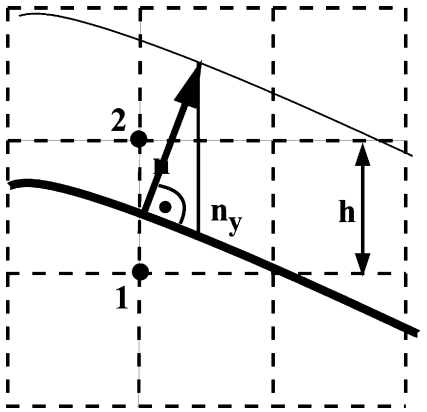

Fig. 2. Extension to initial setting of Adalsteinsson [35]: the difference in the level set values of points 1 and 2 is not assumed to be equal to the grid spacing h but to the quotient h/n_y, where n_y is a projection of the local interface normal n.

the two grid points 1 and 2, but we rescale the distance according to the surface normal as estimated by a central finite difference scheme. This approach ensures that the difference in Φ of the grid points 1 and 2 in Fig. 2 is equal to the quotient h/n_y, where h is the grid spacing and n_y a projection of the local surface normal as shown in Fig. 2. Moreover, we do not implement the fast marching method separately for points inside and outside the front, but instead we use the method based on the initial set of points inside the front to both directions. This ensures that the gradient of Φ converges to one within the first-order accuracy allowed by the fast marching method.

4.3. Implementation

In order to reduce the computational cost of the level set method the computational elements are limited to narrow bands around the interface [3]. This concept is readily implemented in the present method due to

the local support of the underlying particle based functions. The remeshing provides a consistent process by which particles near the interface of the level set are being introduced while particles away from the interface are eliminated.

The equations for the particle locations and volumes (Eq. (3)) are integrated using a Runge–Kutta method of fourth order in all cases. To reduce the computational cost involved with the reconstruction of the level set function from the individual particles (Eq. (16)) we use Linked List [36] and Verlet Lists [37]. The overall cost of the method scales linearly with the number of active particles. For 10^5 particles one time step of the method implemented in FORTRAN 90 requires 1.2 (in 2D) and 2 (in 3D) CPU seconds on an Apple Powerbook with a G4 processor of 1.25 GHz.

5. Results

We present results from the application of the method on a number of benchmark problems in two and three dimensions including pure convection and curvature induced motion. The method is compared quantitatively with existing grid based and hybrid grid-particle level set algorithms as reported in [14,38–40]. The feasibility of the method as a tool for real time cutting procedures in virtual surgery environments is discussed.

5.1. Zalesak's disk and Zalesak's sphere

We demonstrate first the advantages of the proposed method on the rigid body rotation of the Zalesak's disk in a constant vorticity field [41]. Initially, a slotted disk is centered at (0.5, 0.75) with a radius of 0.15, a width of 0.05, and a slot length of 0.25. The velocity field is given by

$$\mathbf{v}(\mathbf{x}) = \frac{\pi}{314}\begin{bmatrix} 0.5 - y \\ x - 0.5 \end{bmatrix}. \tag{30}$$

The disk completes one revolution in the unit domain every 628 time units. This test problem can identify diffusion errors of an interface capturing method. In the proposed method, since the interface is driven by a rigid body rotation there is no particle distortion and hence no need for remeshing.

We present the solution after one revolution for the CF and the SDF description of the level sets, cf. Fig. 3. For the CF approach, the present method requires only 591 particles uniformly distributed within the disk whereas the narrow band of the SDF initialization involves 802 particles. The numerical errors result from the time integration scheme which is of fourth order, and the interpolation kernel used in the initialization, which is of second order. Fig. 4 shows that the error of area computation indeed converges with second order. The number of active particles are listed in Table 1. Note that the numbers for the particles being used along with the mesh in the hybrid methods are not reported [14] and those reported herein represent estimates with an average of 16 particles per cell. The results indicate that the present method requires on the average *five times less computational elements* to achieve the same accuracy as the hybrid particle-level set approach.

As a three-dimensional test case we consider a slotted sphere, corresponding to the two-dimensional Zalesak disk problem. The sphere has a radius of 0.15 and placed at (0.5, 0.75, 0.5) in a unit domain. The slot has a width of 0.05 and length of 0.125. It rotates in the $z = 0.5$ plane around the point (0.5, 0.5, 0.5).

The velocity field describes a rigid body rotation evolving over 628 time units per revolution

$$\mathbf{v}(\mathbf{x}) = \frac{\pi}{314}\begin{bmatrix} 0.5 - y \\ x - 0.5 \\ 0 \end{bmatrix}. \tag{31}$$

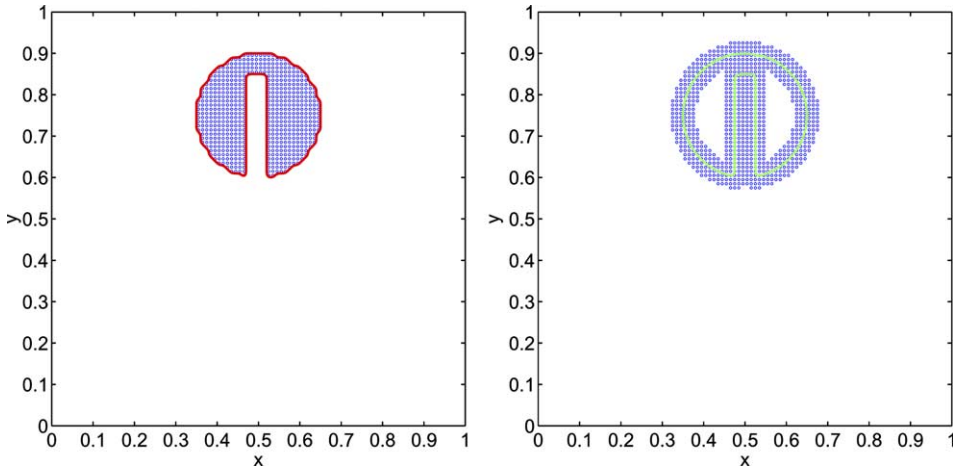

Fig. 3. Zalesak's disk after one revolution. The comparison of level set solutions using color function with particle positions (left, 591 particles) and signed distance function with a narrow band particles (right, 802 particles).

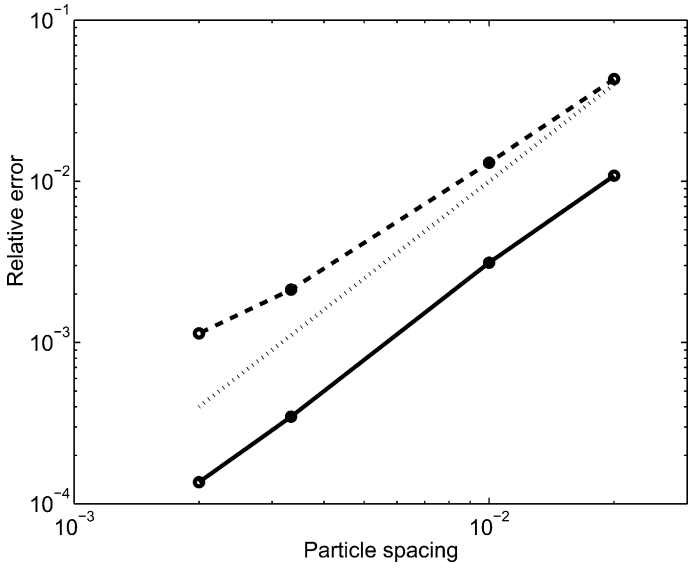

Fig. 4. Comparison of the relative error of the area using color function (dashed line) and level sets (solid line) after one revolution to second convergence (dotted line).

As reported in [14] the level set solution with $100 \times 100 \times 100$ cells (Fig. 5) suffers from numerical diffusion which can be alleviated by the hybrid particle level set method introduced in [14] as shown in Fig. 6.

In the present method the slotted sphere maintains its sharp features (Fig. 7) as the particles follow the rigid body rotation, without any numerical diffusion effects, associated with the advection of the level sets. Fig. 7 shows that the Lagrangian particle level set method performs very well on this problem. Since the particle level set function remains a SDF there is no need for reinitialization.

Table 1
Absolute relative error of the area (exact 0.05822) in the slotted disk case using different Level Set Methods (LSM)

Spacing	Particle LSM		Hybrid particle LSM [14]		LSM [14]
	Error (%)	Particles	Error (%)	Auxiliary particles[a]	Error (%)
Color function					
1/50	4.3	153	14.9	3328	100
1/100	1.3	591	0.3	12,864	5.3
1/300	0.2	5252	–	–	–
1/500	0.1	14,574	–	–	–
Signed distance function					
1/50	1.1	208	14.9	3328	100
1/100	0.3	804	0.3	12,864	5.3
1/300	0.03	7312	–	–	–
1/500	0.01	20,332	–	–	–

[a] Estimated under the assumption of 16 auxiliary particles per narrow band cell.

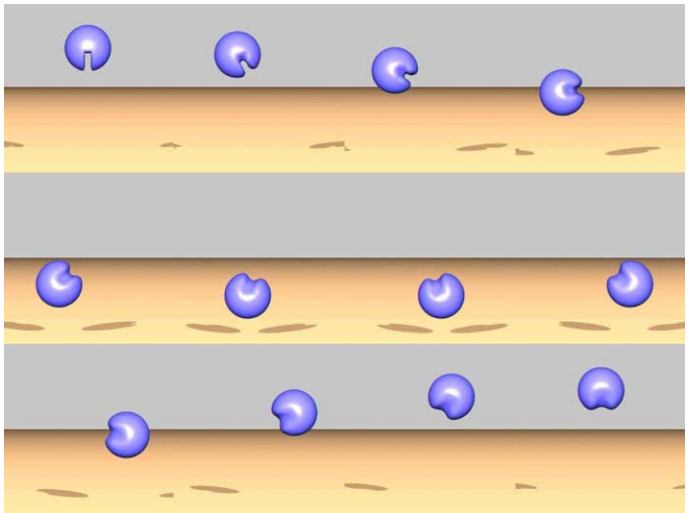

Fig. 5. Zalesak's Sphere: level set solution using $100 \times 100 \times 100$ cells (from [14]).

5.2. Single vortex flow

To test the scheme on resolving thin filaments as they occur in stretching and tearing flows we consider the single vortex flow as introduced by Bell et al. [42]. The initial interface, a circle placed at (0.5, 0.75) with radius 0.15, is shown in Fig. 8 together with the velocity field:

$$\mathbf{v}(\mathbf{x}) = 2\begin{bmatrix} -\sin^2(\pi x)\sin(\pi y)\cos(\pi y) \\ \sin^2(\pi y)\sin(\pi x)\cos(\pi x) \end{bmatrix}. \tag{32}$$

The velocity field stretches the circle into a long filamentary structure which wraps itself around the center of the unit domain. In this case the particles get very distorted by the flow and the remeshing procedure becomes an essential part of the scheme. The threshold for remeshing in our simulations is set to $\Delta H = 10^{-7}$. As a result, the remeshing scheme is applied every time step. Fig. 9 shows the final interface

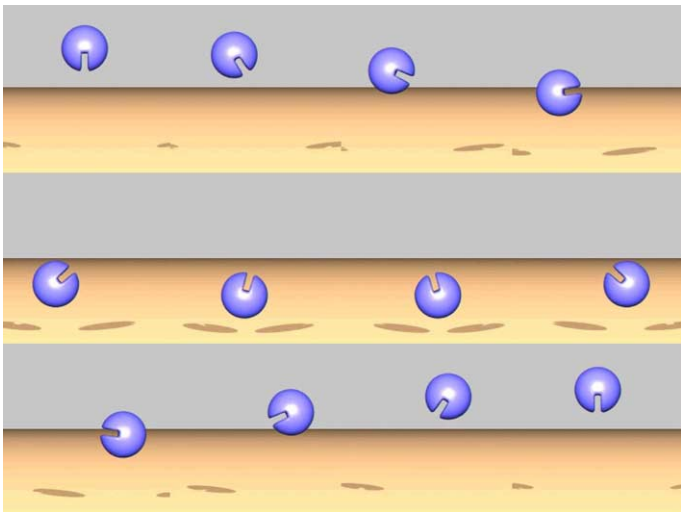

Fig. 6. Zalesak's sphere: particle level set solution of Enright et al. [14] using $100 \times 100 \times 100$ cells and subscale particles.

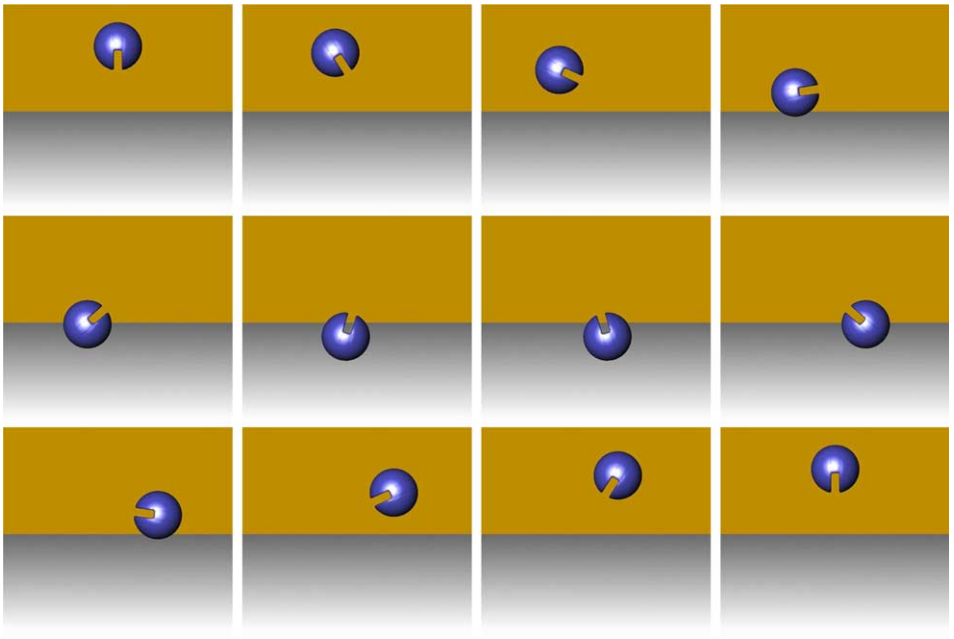

Fig. 7. Zalesak's sphere during one revolution solved by Lagrangian particle level sets (24,351 particles).

at $t = 3$ after 90 time steps when 16,384 particles are used to capture the interface by a CF. When the particle are not remeshed, they soon cease to overlap and the smoothness of the interface gets completely lost and the interface breaks apart as seen in Fig. 9(left). Remeshing ensures that the particle overlap and the interface maintains its smooth features (cf. Fig. 9(right)). Nevertheless in underresolved regions, particles do not describe accurately the level sets and the interface disappears.

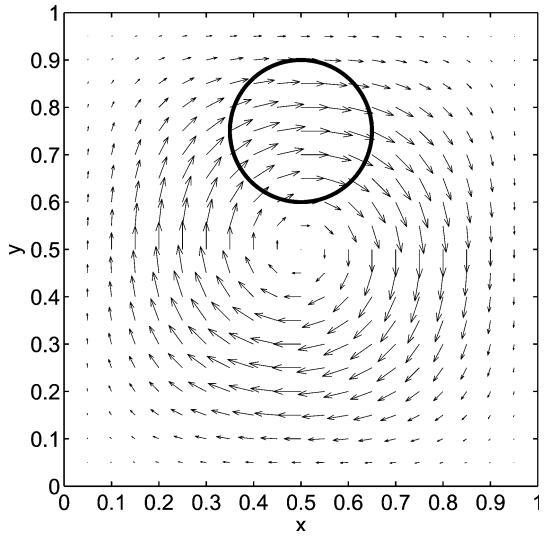

Fig. 8. Single vortex flow: initial interface with velocity field.

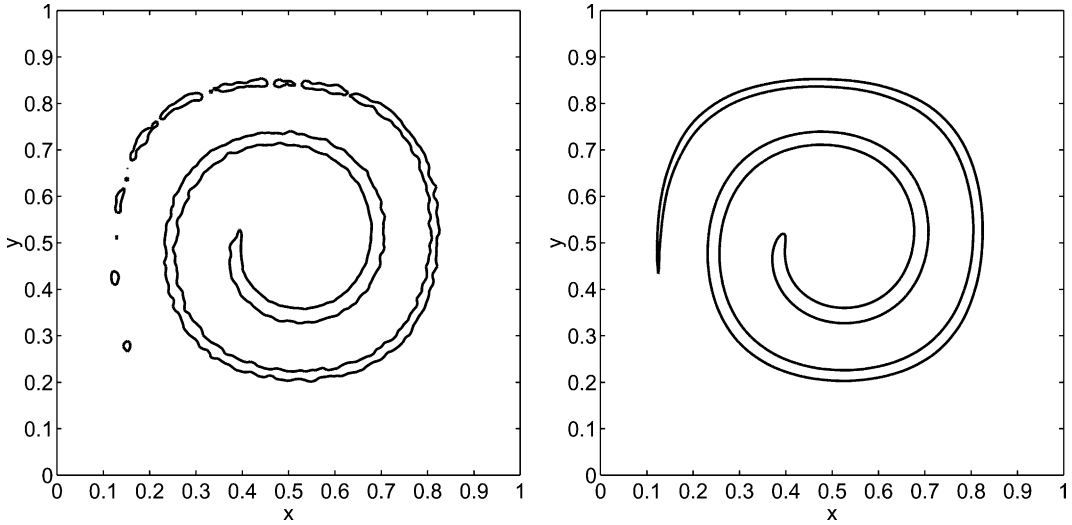

Fig. 9. Zero level set of the single vortex problem at $t = 3$ using Lagrangian particle level sets with a color function having remeshing suppressed (left, 1160 particles) and frequently remeshed (right, 3131 particles).

Using the SDF approach for initialization has similar results (Fig. 10). The remeshing scheme recovers the thin filament very well until the tail becomes underresolved. In both cases, the tail evolution is captured for much longer times when compared to the level set solution obtained by an Eulerian scheme (Fig. 12). Reinitialization is applied every 20 remeshing steps. Our results indicate that the frequency of reinitialization can affect the interface when using the SDF description when applied too often (e.g. at every time step in conjuction with remeshing). When reinitialization is used less often the captured interface is better recovered but the signed distance property of the level set function is not guaranteed any more. Fig. 11 show results when the particle spacing is equal to 10^{-6} and demonstrate that the CF and the SDF approach

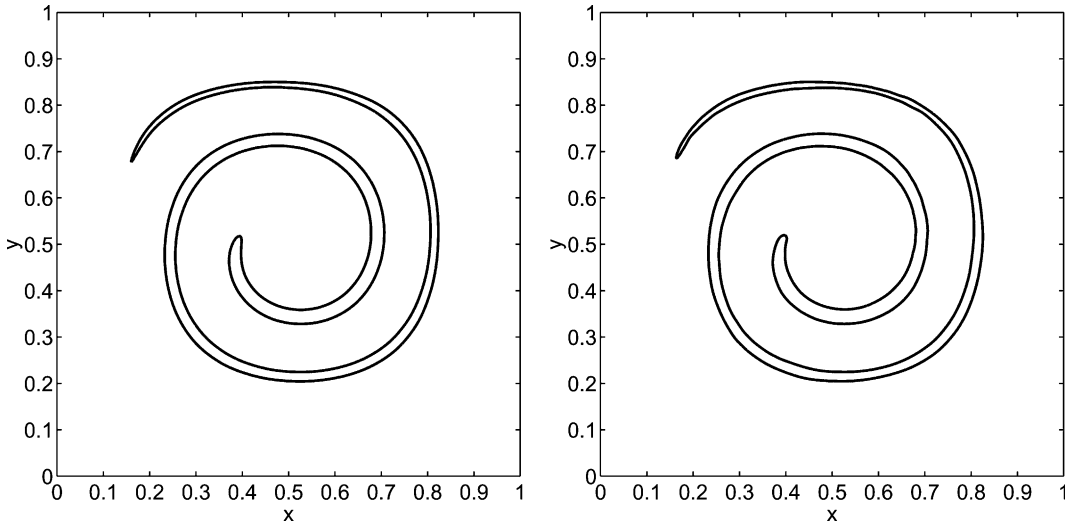

Fig. 10. Zero level set of the single vortex problem at $t = 3$ using Lagrangian particle level sets initialized by a signed distance function using remeshing only (left, 1860 Particles) and remeshing and reinitialization of the signed distance function (right, 4161 particles).

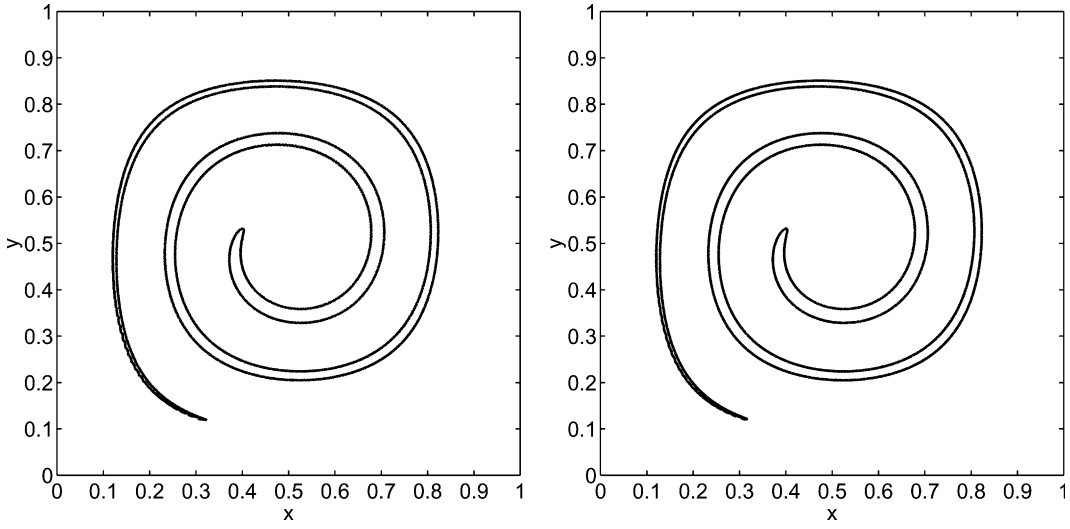

Fig. 11. Zero level set of the single vortex problem at $t = 3$ using Lagrangian particle level sets initialized by a color function (left, 70,946 particles) and a signed distance function (right, 55,914 particles).

converge to the same interface reconstruction. The interface matches nicely with the high resolution front track method as shown in Fig. 12. The quality of the interface is also comparable with the hybrid particle level set method of Enright et al. [14]. In their work they used 128×128 cells along with a number of marker particles seeding randomly the area around the interface. The marker particles were used to track the interface and perform an error correction by providing subgrid resolution. The present Lagrangian particle method is similar to the hybrid particle method as particles are being implemented to adaptively enhance the evolution of the interface. However, the methods differ as in the present work particles are used also to

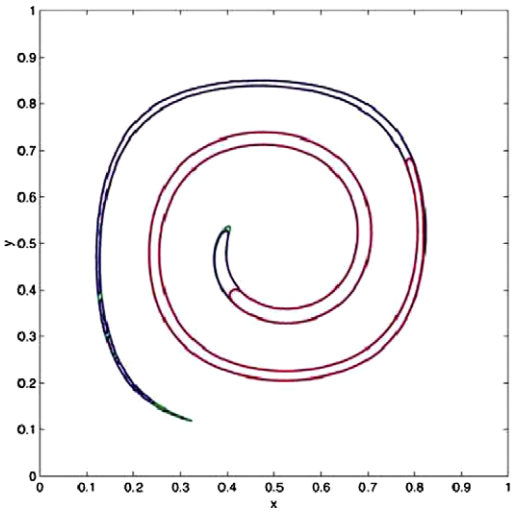

Fig. 12. Zero level set at $t = 3$ from [14]: particle level set method 128×128 cells with subscale particles (blue), Level set method (red), high resolution front track method (green). (For interpretation of the references in colour in this figure legend, the reader is referred to the web version of this article.)

describe the Lagrangian formulation of the level sets. In addition, in the present method, remeshing is utilized to introduce, in a consistent manner, particles in areas where the particle distribution gets non-uniform, in order to maintain the accuracy of the particle description and to regularise the evolution of the particles.

In order to quantify the error of the method the velocity field is reversed by multiplying its value by $\cos(\pi t/T)$, where T is the time of one period. For $T = 8$ the maximal stretched interface is similar to the one in Fig. 11. The final interface at $T = 8$ should match with the initial state. Table 2 shows the errors

Table 2
Absolute relative error of the area (exact 0.07069) in the single vortex problem (240 time steps) using different Level Set Methods (LSM)

Spacing	Particle LSM		Hybrid particle LSM [14]		LSM [14]
	Error	Particles ($t = 0$)	Error	Auxiliary particles[a] ($t = 0$)	Error
Color function					
1/64	9.0	284	1.8	3776	100
1/128	14.2	1160	0.7	15,040	39.8
1/256	6.8	4628	0.4	59,072	10.3
1/1000	1.8	70,688	–	–	–
Signed distance function without reinitialization					
1/64	44.1	236	–	–	–
1/128	2.6	940	–	–	–
1/256	0.8	3692	–	–	–
1/1000	0.003	55,270	–	–	–
Signed distance function with reinitialization					
1/64	17.9	236	1.8	3776	100
1/128	4.2	940	0.7	15,040	39.8
1/256	2.1	3692	0.4	59,072	10.3
1/1000	2.0	56,536	–	–	–

[a] Estimated under the assumption of 16 auxiliary particles per narrow band cell.

of the method in reconstructing the area as compared to the analytical solution. The errors indicate close to second-order convergence when using the SDF without reinitialization and approximately first-order convergence with reinitialization and when using the CF approach (Fig. 13). Fig. 14 shows a selected number of the final interfaces of this problem. Note that the interface reconstruction severely suffers from the reinitialization in simulations using relatively low resolution (i.e. when h is larger than 1/128).

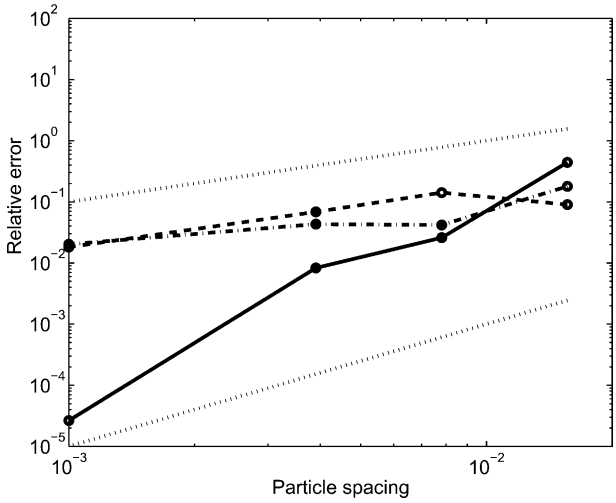

Fig. 13. Single vortex flow: comparison of the relative error of the area at $t = 8$ using CF (dashed line), and SDF level sets without (solid line) and with (dash–dot line) reinitialization. Dotted lines showing first- and second-order scaling.

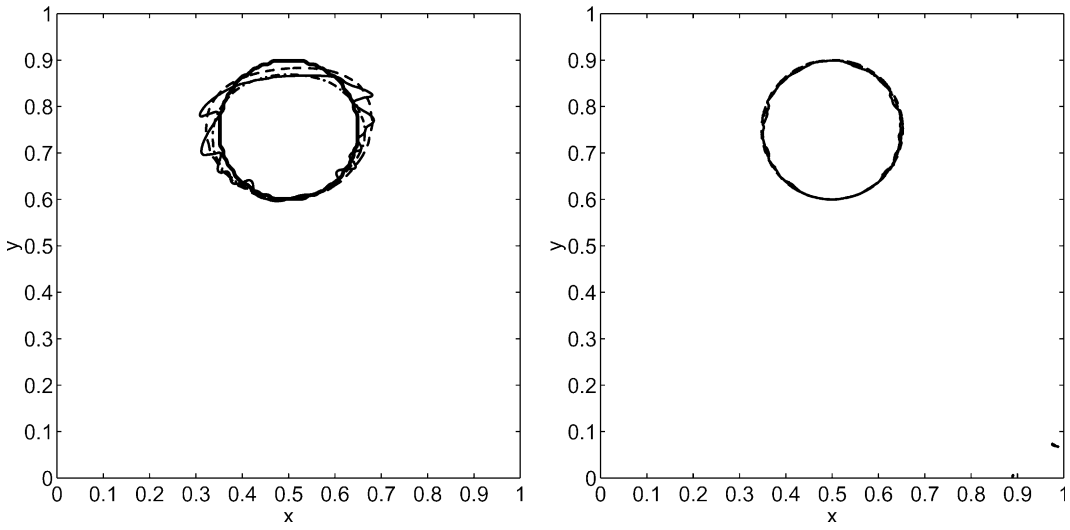

Fig. 14. Time-reversed single vortex problem. Zero level set at $t = 8$ using Lagrangian particle level sets with a particle spacing $h = 1/128$ (left) and $h = 1/1000$ (right) using a color function having remeshing suppressed (thick solid line) and remeshed (dashed line) and using a signed distance function without (dash–dot line) and with reinitialization (thin solid line).

The number of particle varies over time as remeshing creates a new set of particles each time it is applied. The number of particles for the time-reversed single vortex problem typically increases in the first half of the time period as the interface stretches out and decreases in the last third of the time period, cf. Fig. 15.

5.3. Deformation test case in three dimensions

To demonstrate the ability of the method to capture three-dimensional deformations we consider a divergence-free velocity profile proposed by LeVeque [43].

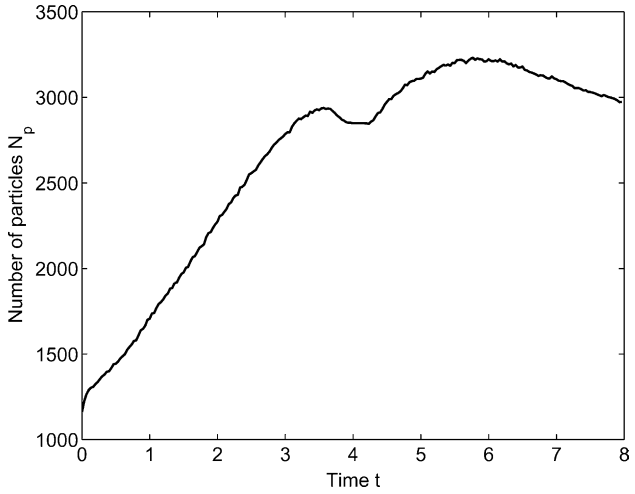

Fig. 15. Time-reversed single vortex problem. The number of particle versus time when using a color function and a particle spacing of $h = 1/128$.

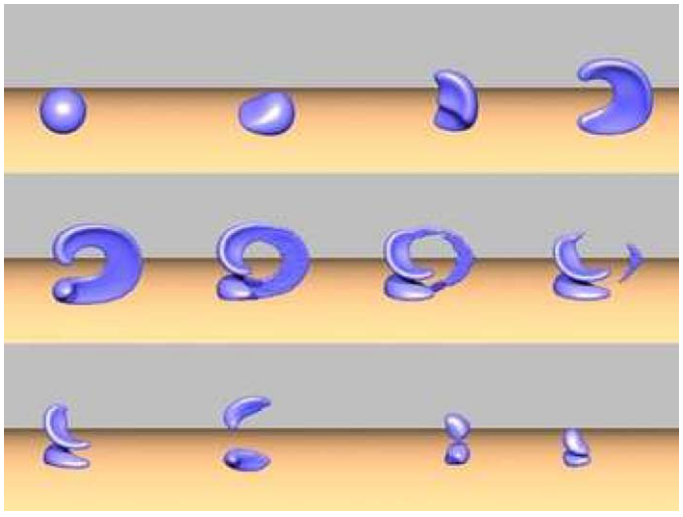

Fig. 16. Deformation test case: level set solution using $100 \times 100 \times 100$ cells from [14].

$$\mathbf{v} = \begin{bmatrix} 2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z) \\ -\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z) \\ -\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z) \end{bmatrix}. \tag{33}$$

A sphere of radius 0.15 is placed within a unit computational domain at (0.35, 0.35, 0.35). At $t = 2$ (after 75 time steps) the velocity field is reversed. Enright et al. [14] presented a level set solution and a hybrid particle level set solution to this problem, for parts of the interface that were underresolved as shown in Figs. 16 and 17. The cell size of these solutions ($\Delta x = 0.01$) corresponds to the particle spacing of one of our simulations. For the same grid size and particle spacing the hybrid method and the present Lagrangian particle method, respectively, do not to resolve the thin interface. Note however that the present method involves a fraction of computational elements when compared to the hybrid method. Both methods give far better results than the purely Eulerian level set method which seems to fail severely on this problem Eq. (16). Figs. 18 and 19 show the Lagrangian particle level set solution with and without remeshing. Initially, particles are initialized using a CF. Particles carrying a color value of less than a small threshold (2.5e−2) are eliminated in further steps. Fig. 18 shows the simulation of the sphere when the particles are undergoing pure advection without been remeshed. In this case the surface quickly becomes very rough and it is ruptured very early. Since the particles are only advected, the final surface after evolving for one period returns perfectly to the initial condition. Remeshing of the particles every second time step improves the smoothness of the surface and delays the rupture as shown in Fig. 19. However, in underresolved regions the interface disappears. The Lagrangian particle level set method is able to recover from this rupture and provides a final smooth surface that shows small spurious features in the $y = 0.5$-plane but overall recovers well the initial condition. The hybrid particle level set method (Fig. 17) results in a better final result as additional subscale particles are utilized, but it still cannot resolve the thin interface at the maximal stretching. Simulations using larger numbers of particles were performed in order to find the resolution that avoids the disappearance of the interface at its maximum stretching. As it is shown in Fig. 20 this is achieved using about 170,000 particles with a spacing that would correspond to an Eulerian mesh of $200 \times 200 \times 200$ for a total of 8,000,000 grid points.

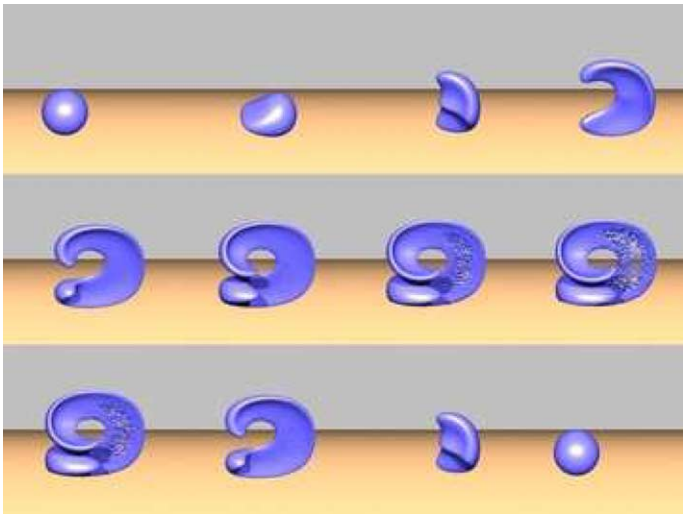

Fig. 17. Deformation test case: particle level set solution of Enright et al. [14] using $100 \times 100 \times 100$ cells and subscale particles.

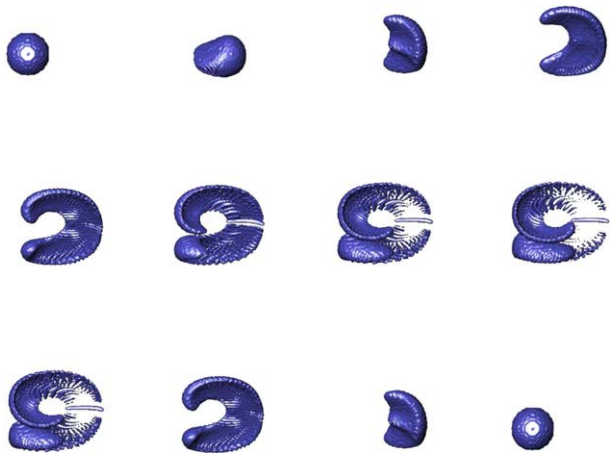

Fig. 18. Deformation test case: particle level set solution without remeshing based on a color function (14,054 particles). Lagrangian particle level set solution *without remeshing* at $t = 0$, $15\Delta s$, $30\Delta s$, $40\Delta s$, $50\Delta s$, $60\Delta s$, $70\Delta s$, $75\Delta s$, $80\Delta s$, $100\Delta s$, $120\Delta s$ $150\Delta s$, where $\Delta s = 2/150$.

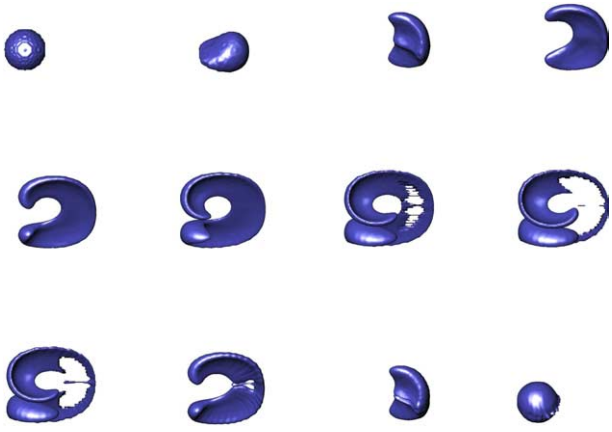

Fig. 19. Deformation test case: Lagrangian particle level set solution based on a color function (minimum 14,054 particles, maximum 30,739 particles). Lagrangian particle level set solution *with remeshing* at $t = 0$, $15\Delta s$, $30\Delta s$, $40\Delta s$, $50\Delta s$, $60\Delta s$, $70\Delta s$, $75\Delta s$, $80\Delta s$, $100\Delta s$, $120\Delta s$ $150\Delta s$, where $\Delta s = 2/150$.

5.4. Flow under mean curvature

The particle description of the Level Set can be extended to advection determined by mean curvature of the interface as

$$\Phi_t - \kappa|\nabla\Phi| = 0, \tag{34}$$

where κ is the curvature as defined by Eq. (8). The computation of spatial derivatives on particle locations makes use of the approximations described in Eqs. (21) and (22). We determine the accuracy of the curvature evaluation by computing the curvature of an ellisoid with main axes' length of $a = 0.1$, $b = 0.2$, $c = 0.2$

and comparing it to the analytical solution. The L_∞ norm of error is shown in Fig. 21. The error of the SDF description converges with the second order of the interpolation kernel while the CF description is less accurate.

To investigate the accuracy of our method in resolving interface normals (Eq. (7)), we present the L_∞-error of the normal at a circular interface with radius R in terms of R/ϵ, where ϵ is the characteristic length of the kernel (Fig. 22). We observe close to second order convergence for the SDF description while sub-linear convergence is observed for the CF description.

In order to further validate our method on the computation of normals, we implement the SDF description to track an anisotropic evolution of an interface as proposed by Sethian and Strain [44]. The anisotropic speed function u_A of this test case is defined by

$$u_A = 1 - \epsilon_C(1 - \cos(k_A(\theta + \theta_0)))\kappa, \tag{35}$$

where κ is the curvature and ϵ_C is set to 0.25 and $k_A = 4$. Fig. 23 shows the interface evolution for the phase angle $\theta_0 = 0$ (left) and $\theta_0 = \pi/4$ (right) using about 3000 particles. In both cases, the speed function transforms the circle into a square. In the first case, the preferred modes of growth are along the x- and y-axes, whereas in the second case the preferred modes of growth are along the diagonals. The corners turn out sharper in the first case. These results compare well with the results of Sethian and Strain [44] using finite difference schemes.

We simulate the front evolution of an H-shape contour using Runge–Kutta fourth order with a time step of 10^{-4} and a particle spacing of 10^{-2}. Grayson [45] showed that all simple close curves flowing under

Fig. 20. Effect of the resolution on the surface at $t = 1$: $h = 1/100$, 29,073 particles (left), $h = 1/160$, 94,033 particles (middle), and $h = 1/200$, 169,500 particles (right).

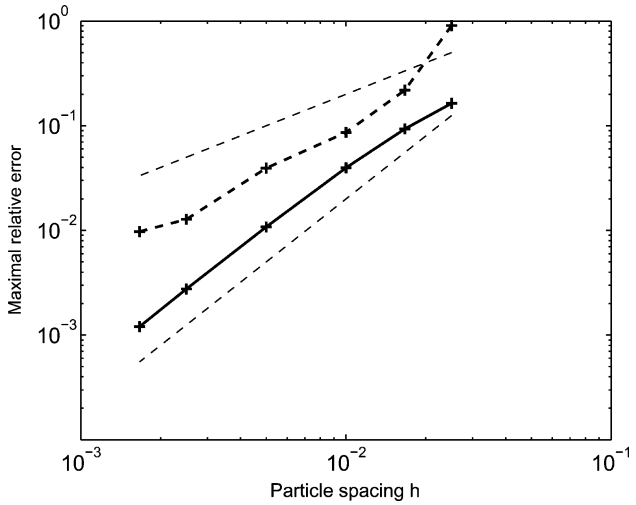

Fig. 21. Convergence of the calculation of mean curvature for the surface of ellipsoid based on the CF (dashed line) and SDF (bold line) descriptions, along with the lines of first- and second-order convergence.

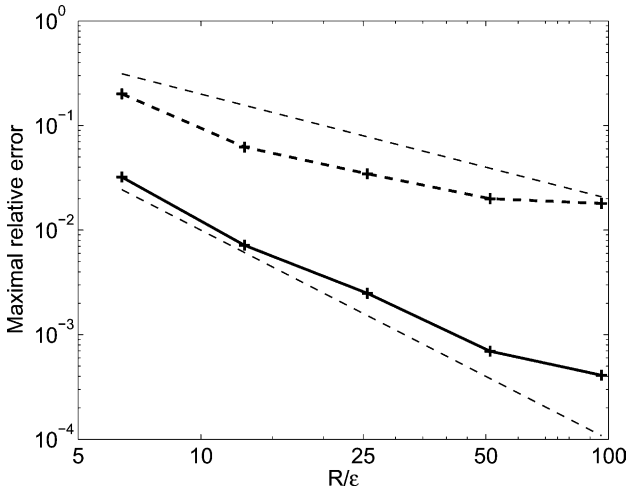

Fig. 22. Convergence of the interface normals computed on a circle for the SDF (solid line with markers) and CF description (dashed line with markers), along with the lines (dashed) of first- and second-order convergence.

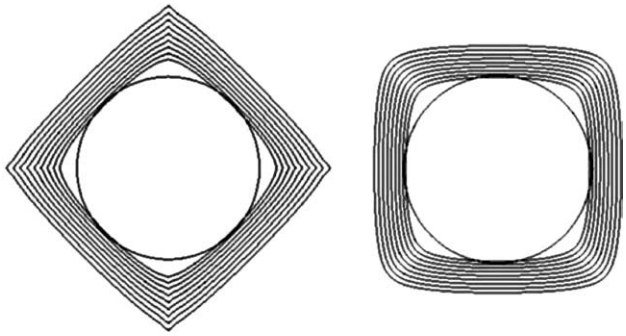

Fig. 23. Interface evolution under a fourfold anisotropic speed function with phase angle $\theta_0 = 0$ (left) and $\theta_0 = \pi/4$ (right) using about 3000 particles.

curvature shrink to a point. The H-shaped contour is progressively becoming a shrinking ellipsoid. We compare with the results of an Adaptive Mesh Redistribution (AMR) method [38] in Fig. 24. The effective resolution of the AMR is comparable to the resolution of the particle simulation. The two solutions match very well in the first 50 time steps and up to $t = 5 \times 10^{-3}$. The interfaces differ slightly at later time steps after the interface velocity has decayed significantly. As the analytical solution for this problem is not available we cannot judge which solution is more accurate.

To demonstrate the performance in three dimensions we simulate the collapse of a dumbbell that is a well known curvature flow example [39,40] as it exposes a singularity. The mean curvature flow pinches off the handle that separates into two pieces, which continue to shrink and finally vanish. Grayson [46] used this example to show that non-convex shapes in three dimensions may in fact not shrink to one sphere. The dumbbell is made up of two spheres, each of radius 0.3, and connected by a cylindrical handle of radius 0.15. The x-axis is the axis of symmetry. We choose a particle spacing of $0.09\overline{7}$

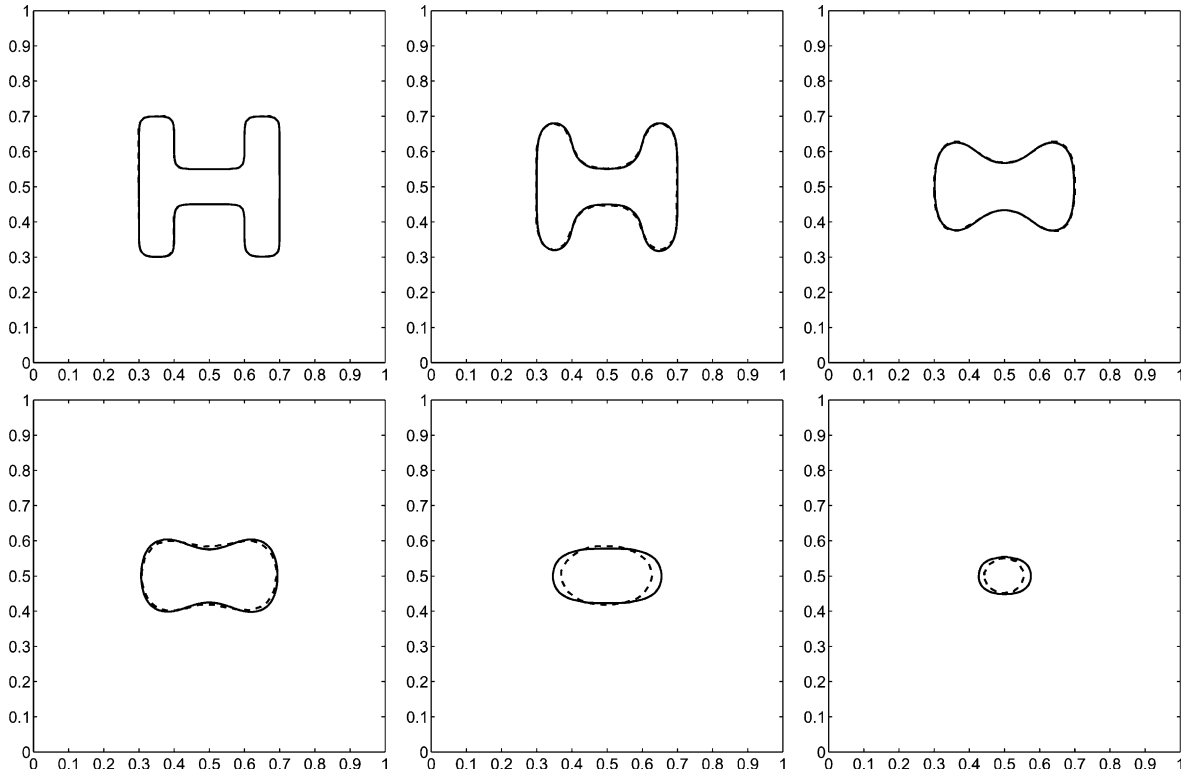

Fig. 24. Curvature flow of an H-shaped contour at $t = 0$, 10^{-3}, 3×10^{-3}, 5×10^{-3}, 10×10^{-3}, 14×10^{-3} with initially 1701 particles (solid line) in comparison to Tang [38] (dashed line).

and a time step of 2×10^{-5}. The particles are reinitialized every 10th time step. Fig. 25 shows the surface as it appears initially, after shrinkage, when reaching the singularity and after the break up. The quality of the results is comparable with the finite difference solution of Sethian [39,40] as seen in Fig. 26. The resolution of domain and the size of the time step are equivalent in both simulations. The interface is plotted every 100 time steps, later, when approaching the singularity, it is plotted every 10 time steps.

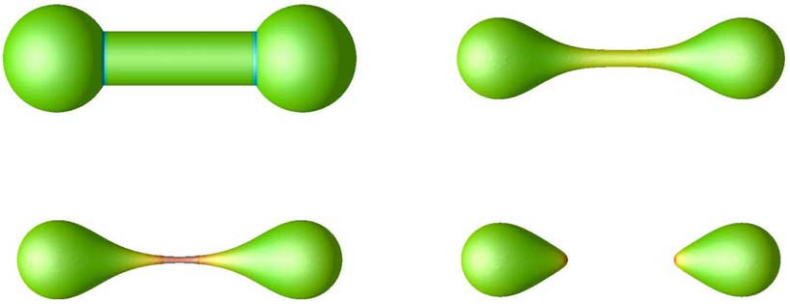

Fig. 25. Evolution of a dumbbell shaped surface under mean curvature flow at $t = 0$, 10^{-3}, 1.1×10^{-3}, 1.26×10^{-3} (initially 334,616 particles).

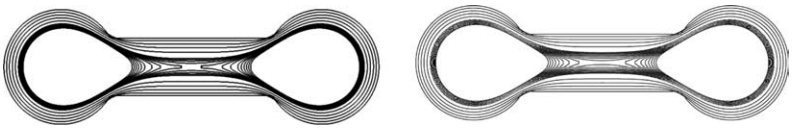

Fig. 26. Cross sections of the evolving dumbbell (left) in comparison to Chopp and Sethian [39] (right).

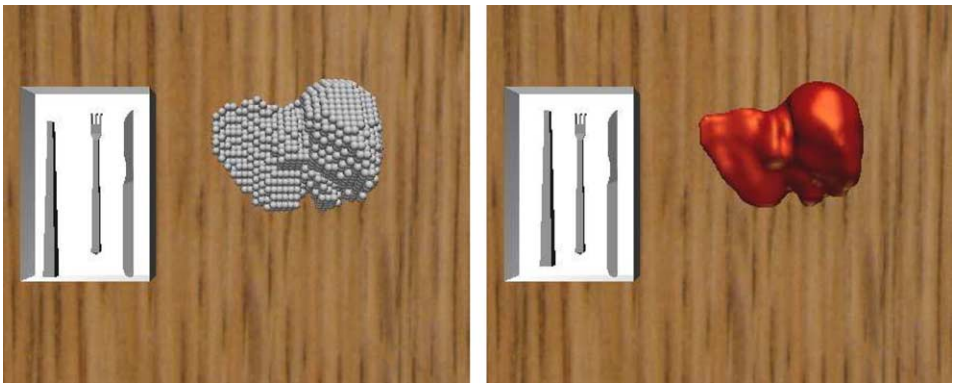

Fig. 27. Liver topology reconstruction using 3209 particles.

5.5. Virtual cutting using Lagrangian particle level sets

The simplicity and efficiency of the proposed method enables simulations associated with virtual cutting of soft biological tissue.

We consider a liver topology that was segmented from image data of the Visible Human Project. Based on the a triangular mesh of the topology, particles are placed inside the liver surface and they are assigned values following a CF approach. Fig. 27 shows the surface reconstruction of the liver based on 3209 particles. In order to simulate cutting, whenever a medical device collides with one of the particles inside, the contribution of this particles it is removed from the superposition of Eq. (16). Hence the reconstruction of the surface is computationally very inexpensive as the new surface is reconstructed according to:

$$\Phi(\mathbf{x})^{\text{new}} = \Phi(\mathbf{x})^{\text{old}} - \sum_{q}^{M} v_q \Phi_q \zeta_\epsilon(\mathbf{x} - \mathbf{x}_q), \tag{36}$$

where M denotes the (small) number of particles detected during the collision process. The collision detection is performed by a collision detection library for deformable objects provided by Heidelberger et al. [47]. This algorithm shows high efficiency and enables interactive simulations (Fig. 28) when the devices moved into the liver are not thinner than the particle spacing. Adaptive insertion of particles having smaller core size is necessary in order to refine this process.

6. Summary and conclusions

We have presented a novel Lagrangian particle level set method for capturing interfaces evolving under pure advection and curvature induced motion in two and three dimensions. The method is adaptive as particles adapt to resolve the evolution of the level sets and a consistent remeshing procedure is employed in

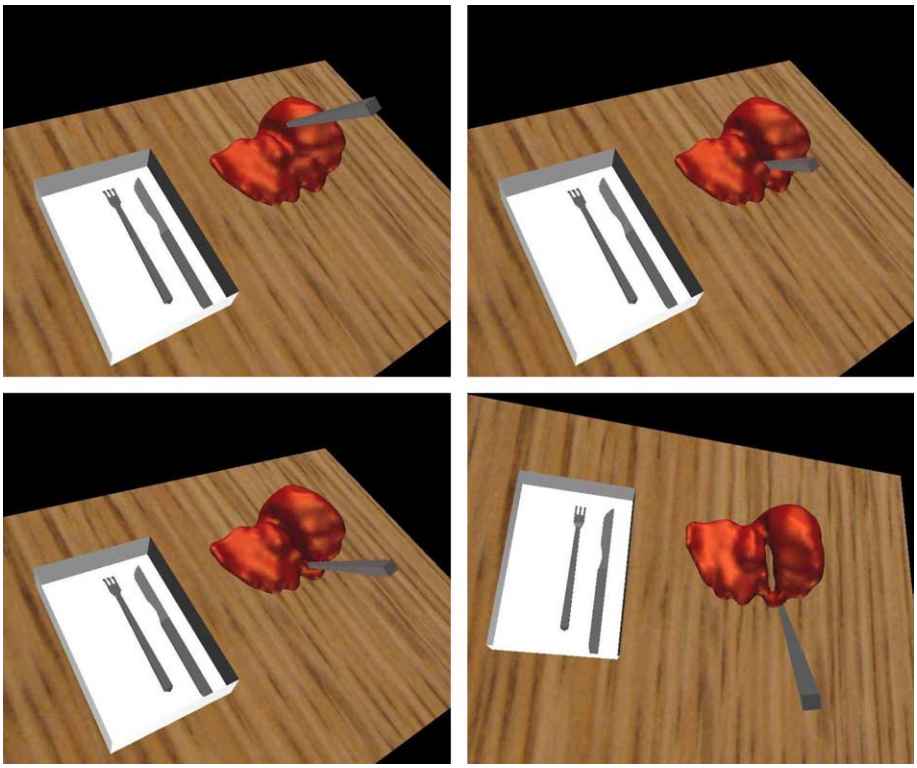

Fig. 28. Particles assigned with a color functions are removed from the superposition in real time when hit by an instrument.

order to ensure the convergence of the method when the particles get distorted by the flow map. Remeshing enables also the use of fast marching methods for the reinitialization of the level sets.

The present Lagrangian particle method provides a bridging description of interface tracking and interface capturing methodologies as it is adaptive and at the same time involves an implicit description of the interface. The efficiency and accuracy of the method, as well as comparison with related methodologies, is demonstrated in a number of two- and three-dimensional benchmark problems. The method is shown to be well capable in tracking curvature induced motion as well as in resolving the sharpness of corners and their respective normals as they occur in semiconductor simulations for etching and deposition. The simplicity of the method in reconstructing interfaces by a linear particle superposition makes it highly suitable for rapid tracking of large deformations and cutting of interfaces in virtual surgery environments where fast simulations of cutting of biological tissue are necessary. Present work involves the development of a multiscale Lagrangian particle level set technique by considering spatially varying particle mollifications [48,49] and the coupling of these methods to atomistic simulations as they pertain to micro- and nano-manufacturing problems.

Acknowledgments

We wish to thank Jens Walther, Michael Bergdorf and Pierre Gautier (Institute of Computational Science, ETHZ) for several helpful discussions and Sidclei da Silva for his contributions to the Virtual Surgery

project. This project was funded by the Swiss National Science Foundation, NCCR 'Computer Aided and Image Guided Medical Interventions' (COME).

References

[1] C.W. Hirt, B.D. Nichols, Volume of fluid (Vof) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1) (1981) 201–225.
[2] S. Osher, J.A. Sethian, Front propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulation, J. Comput. Phys. 79 (1) (1988) 12–49.
[3] J.A. Sethian, A fast marching level set method for monotonically advancing fronts, Proc. Natl. Acad. Sci. USA 93 (4) (1996) 1591–1595.
[4] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annu. Rev. Fluid Mech. 31 (1999) 567–603.
[5] J.A. Sethian, Evolution, implementation, and application of level set and fast marching methods for advancing fronts, J. Comput. Phys. 169 (2) (2001) 503–555.
[6] S. Osher, R. Fedkiw, The Level Set Method and Dynamics Implicit Surfaces, Springer-Verlag, New York, 2002.
[7] J.A. Sethian, Level Set Methods and Fast Marching Methods, Cambridge University Press, Cambridge, UK, 1999.
[8] S. Osher, R.P. Fedkiw, Level set methods: an overview and some recent results, J. Comput. Phys. 169 (2) (2001) 463–502.
[9] F. Gibou, R. Fedkiw, R. Caflisch, S. Osher, A level set approach for the numerical simulation of dendritic growth, J. Sci. Comput. 19 (1–3) (2003) 183–199.
[10] J.A. Sethian, P. Smereka, Level set methods for fluid interfaces, Annu. Rev. Fluid Mech. 35 (2003) 341–372.
[11] S. Bryson, D. Levy, High-order central WENO schemes for multidimensional Hamilton–Jacobi equations, SIAM J. Numer. Anal. 41 (4) (2003) 1339–1369.
[12] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, J. Comput. Phys. 141 (1998) 112–152.
[13] J. Strain, A fast semi-lagrangian contouring method for moving interfaces, J. Comput. Phys. 161 (2) (2001) 512–536.
[14] D. Enright, R. Fedkiw, J. Ferziger, I. Mitchell, A hybrid particle level set method for improved interface capturing, J. Comput. Phys. 183 (1) (2002) 83–116.
[15] L. Rosenhead, The formation of vortices from a surface of discontinuity, Proc. R. Soc. Lond. A 134 (1931) 170–192.
[16] A.J. Chorin, Numerical study of slightly viscous flow, J. Fluid Mech. 57 (4) (1973) 785–796.
[17] N. Zabusky, M. Hughes, K. Roberts, Contour dynamics for the Euler equations in two dimensions, J. Comput. Phys. 30 (1979) 96–106.
[18] G.-H. Cottet, P. Koumoutsakos, Vortex Methods – Theory and Practice, Cambridge University Press, New York, 2000.
[19] P. Koumoutsakos, A. Leonard, High-resolution simulation of the flow around an impulsively started cylinder using vortex methods, J. Fluid Mech. 296 (1995) 1–38.
[20] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible 2-phase flow, J. Comput. Phys. 114 (1) (1994) 146–159.
[21] J.A. Sethian, Fast marching methods, SIAM Rev. 41 (2) (1999) 199–235.
[22] J.P. Morris, Simulating surface tension with smoothed particle hydrodynamics, Int. J. Numer. Methods Fluids 33 (3) (2000) 333–353.
[23] O.H. Hald, Convergence of vortex methods for Euler's equations III, SIAM J. Numer. Anal. 24 (3) (1987) 538–582.
[24] J.T. Beale, A convergent 3-D vortex method with grid-free stretching, Math. Comput. 46 (1986) 401–424.
[25] J.J. Monaghan, Smoothed particle hydrodynamics, Annu. Rev. Astron. Astrophys. 30 (1992) 543–574.
[26] J.D. Eldredge, A. Leonard, T. Colonius, A general deterministic treatment of derivatives in particle methods, J. Comput. Phys. 180 (2) (2002) 686–709.
[27] P. Degond, S. Mas-Gallic, The weighted particle method for convection–diffusion equations. Part 1: The case of an isotropic viscosity, Math. Comput. 53 (188) (1989) 485–507.
[28] G.-H. Cottet, B. Michaux, S. Ossia, G. VanderLinden, A comparison of spectral and vortex methods in three-dimensional incompressible flows, J. Comput. Phys. 175 (2002) 702–712.
[29] G.-H. Cottet, Artificial viscosity models for vortex and particle methods, J. Comput. Phys. 127 (1996) 299–308.
[30] P. Koumoutsakos, Vorticity flux control in a turbulent channel flow, Phys. Fluids 11 (2) (1999) 248–250.
[31] D.L. Chopp, Computing minimal-surfaces via level set curvature flow, J. Comput. Phys. 106 (1) (1993) 77–91.
[32] M. Sussman, E. Fatemi, An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow, SIAM J. Sci. Comput. 20 (4) (1999) 1165–1191.
[33] G.-H. Cottet, E. Maitre, A level-set formulation for fluid-structure interactions with immersed surfaces, Math. Models and Methods in Appl. Sciences, accepted.

[34] B. Engquist, A.-K. Tornberg, R. Tsai, Discretization of Dirac delta functions in level set methods, J. Comput. Phys. 207 (1) (2004).
[35] D. Adalsteinsson, J.A. Sethian, A fast level set method for propagating interfaces, J. Comput. Phys. 118 (2) (1995) 269–277.
[36] R.W. Hockney, J.W. Eastwood, Computer Simulation Using Particles, second ed., IOP, 1988.
[37] L. Verlet, Computer experiments on classical fluids. I. Thermodynamical properties of Lennard–Jones molecules, Phys. Rev. 159 (1) (1967) 98–103.
[38] H.-Z. Tang, T. Tang, P. Zhang, An adaptive mesh redistribution method for nonlinear Hamiltion–Jacobi equations in two- and three-dimensions, J. Comput. Phys. 188 (2003) 543–572.
[39] D. Chopp, J. Sethian, Flow under curvature: singularity formation, minimal surfaces, and geodesics, Exp. Math. 2 (4) (1993) 235–255.
[40] J. Sethian, Numerical algorithms for propagating interfaces: Hamilton–Jacobi equations and conservations laws, J. Differential Geometry 31 (1990) 131–161.
[41] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 31 (3) (1979) 335–362.
[42] J.B. Bell, P. Colella, H.M. Glaz, A 2nd-order projection method for the incompressible Navier–Stokes equations, J. Comput. Phys. 85 (2) (1989) 257–283.
[43] R.J. LeVeque, High-resolution conservative algorithms for advection in incompressible flow, SIAM J. Numer. Anal. 33 (2) (1996) 627–665.
[44] J. Sethian, J. Strain, Crystal growth and dendritic solidification, J. Comput. Phys. 98 (1992) 231–253.
[45] M. Grayson, The heat equation shrinks embedded plane curves to round points, J. Diff. Geom. 26 (1987) 555–558.
[46] M. Grayson, A short note on the evolution of surfaces via mean curvatures, J. Diff. Geom. 58 (1989) 285–314.
[47] B. Heidelberger, M. Teschner, M. Gross, Real-time volumetric intersections of deforming objects, Vision Modeling and Visualization, IOS Press, 2003, pp. 461–468.
[48] M. Bergdorf, G.-H. Cottet, P. Koumoutsakos, Multilevel adaptive particle methods for convection–diffusion equations, SIAM Multiscale Model. Simulat., in press.
[49] P. Koumoutsakos, Multiscale flow simulations using particles, Annu. Rev. Fluid Mech. 37 (2005) 457–487.