Computers and Fluids 168 (2018) 232–244

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

Benchmark solutions

An advection velocity correction scheme for interface tracking using the level-set method

Kiran Satheesh, Gowtham Srinivas*, Santosh Hemchandra

Department of Aerospace Engineering, Indian Institute of Science, Bangalore 560012, India

ARTICLE INFO

Article history:
Received 18 July 2017
Revised 11 March 2018
Accepted 6 April 2018
Available online 7 April 2018

Keywords:
Level-set
Re-initialization
Sub-cell fix
Advection velocity correction

ABSTRACT

We present an advection velocity correction (AVC) scheme for interface tracking using the level-set method in this paper. The key idea is to apply a correction to the interface advection velocity at points adjacent to the zero level-set, so as to enforce the preservation of the signed distance function property at these points. As such, the AVC scheme eliminates the need for explicit sub-cell fix approaches as reinitialization at points adjacent to the zero level-set is not needed. This approach of correcting the advection velocity field near the interface and computing the signed distance function; SDF to a high order of accuracy near the interface, rather than applying an explicit sub-cell fix during the reinitialization step represents the key novel aspect of the AVC scheme. We present results from using the AVC scheme along with advection and reinitialization schemes using upwind finite differencing on uniform meshes in this paper. These results are determined for four canonical test problems: slotted disk rotation, deforming sphere, interacting circles and vortex in a box. We compare these results with corresponding results determined using a recently proposed explicit sub-cell fix based reinitialization scheme (CR2). These comparisons show that the AVC scheme yields significantly improved conservation of enclosed volume/area within the interface. Note, the present AVC scheme achieves this by *only* modifying velocity field values at mesh points. Therefore, the AVC algorithm can in principle be used within the framework of nearly any numerical scheme used to compute interface evolution using the level-set method, even on non-uniform and unstructured meshes, in order to achieve improvements in solution quality.

1. Introduction

Interface tracking is a key problem which arises in many application areas of computational modeling [1,2] which involves tracking the evolution of a closed interface (i.e. the boundary of some open set in the computational domain), as shown schematically by the red curve in Fig. 1, under the influence of a velocity field $\vec{v}(\vec{x},t)$. Interfaces are represented in the level-set method by constructing a higher dimensional function $\phi(\vec{x},t)$ such that, $\phi(\vec{x},t) = 0$ for all points that lie on the interface, $\phi(\vec{x},t) < 0$ at points inside the interface and $\phi(\vec{x},t) > 0$ at points outside the interface - see Fig. 1. As such, the shape of the zero level-set of ϕ corresponds to the shape of the interface. Further, ϕ must vary monotonically near the interface and may not have an extremum or critical point at a point on the interface. Thus, the evolution of the interface under the action of $\vec{v}(\vec{x},t)$, is implicitly tracked by requiring that the value of ϕ at points on the interface remain unchanged. This yields the level-set advection equation as follows (see for e.g. [1]),

$$\frac{\partial \phi}{\partial t} + \vec{v} \cdot \nabla \phi = 0 \tag{1}$$

where, $\vec{v}$ may be defined either only on the interface or at all points in the domain, depending on the specific details of the physics governing the evolution of the interface.

The implicit representation of the interface using ϕ in the level-set method means that it can be applied to interfaces that are either simply connected or multiply connected. This is because the function ϕ is well defined and continuous in both these cases. Also, topological changes in the interface during the course of its evolution are automatically captured by solving Eq. (1). This is the significant advantage in terms of simplicity of implementation of the level-set method over explicit methods for interface evolution such as particle tracking.

Eq. (1) is numerically solved using techniques developed for advection equations or Hamilton–Jacobi equations (see e.g. refs. [1,3]), starting from an initial condition for ϕ whose zero level-set coincides with the shape of the interface. It is common to choose the value of ϕ at every point in the domain to be the minimum signed distance of the point from the interface. The sign of ϕ is chosen

* Corresponding author.
E-mail address: gowthams@iisc.ac.in (G. Srinivas).

https://doi.org/10.1016/j.compfluid.2018.04.010
0045-7930/

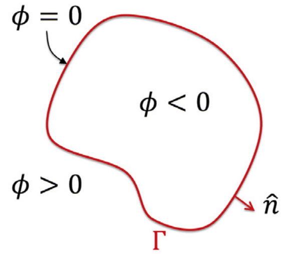

Fig. 1. Schematic showing the representation of an interface in the framework of the level-set method. The closed red curve shows the interface, Γ.(For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

as negative when the point lies inside the interface and positive, when it lies outside [2]. The resulting signed distance function (SDF) is a continuous function that nominally satisfies $|\nabla\phi| = 1$ (SDF property) everywhere. This ensures that numerical solutions to Eq. (1) are accurately computed, as long as, ϕ retains the SDF property during the computation. However, in general, $\vec{v}$ can potentially cause steep gradients to develop in ϕ under the action of Eq. (1). This in turn, results in large numerical errors in the computation of ϕ and as a consequence, a loss of accuracy in the shape of its zero level-set.

Two classes of techniques have been developed to address this problem. The first is the fast marching technique of Adalsteinsson and Sethian [4] where, $\vec{v}$ is extended off the zero level-set of ϕ along the orthogonal trajectories of its level-sets such that the value of $\vec{v}$ along these trajectories does not change. This is sufficient to maintain the SDF property of ϕ throughout the computation. The other class of techniques is based on the re-initialization approach where the values of ϕ at points away from the zero level-set are reset to the value of the SDF from the latter, by solving an auxiliary partial differential equation as follows [5].

$$\frac{\partial\phi}{\partial\tau} + S(\phi^0)(|\nabla\phi| - 1) = 0 \tag{2}$$

where, ϕ^0 is the level-set field resulting from solving Eq. (1) through one or more time steps and $S(x)$ is the sign function which takes values 1, 0 and -1 for positive, negative and zero values of its argument. Eq. (2) is solved after one or more time steps of Eq. (1), which at steady state yields, $|\nabla\phi| = 1$ at points where $\phi \neq 0$. The pseudo-time, τ, is introduced to anticipate the use of numerical schemes based on time stepping methods to solve Eq. (2), using ϕ^0 as an initial condition. Also, the second term on the left of Eq. (2) formally disappears at points where $\phi^0 = 0$. Therefore, Eq. (2) nominally leaves the shape of the zero level-set unchanged.

Errors resulting from the numerical discretization of Eq. (2) at points adjacent to the zero level-set of ϕ^0, cause spurious movement of the latter resulting in inaccurate interface evolution. This can cause violation of mass conservation when the level-set method is used in conjunction with the Navier–Stokes equations for fluid flow problems. Several approaches have been proposed to mitigate this problem such as, using high order schemes [6,7] and coupling the level-set method with other types of interface tracking such as the volume of fluid (VOF) based approaches [8,9]. Russo and Smereka [10] show that Eq. (2) can be rewritten as an advection equation with an advection velocity of unity, directed along the orthogonal trajectories of ϕ. As such, numerical errors arise at points adjacent to the interface when upwind finite difference schemes are used to solve Eq. (2) because of incorrect upwind discretizations at these points. They proposed a sub-cell fix approach where, the upwind discretization of spatial derivatives at points adjacent to the zero level-set of ϕ^0 are replaced with those computed using a direct estimate of their distance from the zero level-set [10]. Alternatively, they have suggested that the numerical scheme at these points may be bypassed and the value of ϕ be set to the sub-cell fix estimate.

Recently, Hartmann et al. [11] have developed two schemes that improve upon the baseline sub-cell fix distance estimate proposed by Russo and Smereka [10]. The CR1 [11] scheme derives a sub-cell fix distance estimate at points adjacent to the zero level-set by formally minimizing the deviations of ϕ from the SDF property as well as spurious movement of the zero-level set introduced by the sub-cell fix of Russo and Smereka [10] in a least squares sense. The CR2 [11] scheme enforces a constraint on the sub-cell fix estimate so as to explicitly anchor the location of the zero-level set within a grid cell. In a companion paper, Hartmann et al. [12] have generalized these ideas to be amenable to higher order spatial discretization. Both of these approaches have yielded significant improvements in solution quality over the original proposition of Russo and Smereka [10].

We present a different technique for computing values of ϕ at grid points adjacent to the interface in this paper, motivated by the principle underlying the velocity extension approach of Adalsteinsson and Sethian [4]. They show that the SDF property is preserved if the component of $\vec{v}$ along the orthogonal trajectories to the level-sets of ϕ is invariant. Our technique enforces this property in a discrete sense at grid points adjacent to the zero level set by applying a correction to $\vec{v}$ at grid points adjacent to the zero level-set of ϕ. This velocity correction ensures that the level-sets of ϕ passing through these grid points, move with the same velocity normal to themselves as the zero level-set. Thus, this ensures that the SDF property is preserved over an advection timestep at grid points adjacent to the zero level-set of ϕ. Therefore, Eq. (2) is not solved at these points and no additional sub-cell fix is applied either. As such, the values of ϕ at these points serve as a Dirichlet boundary condition for the remaining points in the domain where Eq. (2) is solved to restore the SDF property of ϕ. We call this scheme the advection velocity correction (AVC) scheme. To the best of the authors' knowledge, this approach for preserving the SDF property of the level-set function at points near the interface by correcting the advection velocity field has not been proposed earlier and represents the key novel contribution of this paper.

We compare results from the AVC scheme to those obtained using the CR2 scheme of Hartmann et al. [11] which is chosen as a reference method. The same numerical schemes are used in both AVC and CR2 computations for level-set advection and reinitialization. The key difference between the two sets of computations is that the CR2 scheme explicitly computes the SDF at grid points adjacent to its zero level-set using a sub-cell fix approach during the reinitialization step [11], while, the AVC scheme achieves the same result by correcting advection velocities at these points in the advection step. Test cases that are representative of interface evolution scenarios that can arise in interface tracking problems are chosen for comparison. All of these computations are performed on a mesh with uniform grid spacing in all three directions. The results show that for a given mesh size, the AVC scheme provides results that are significantly improved with respect to area and volume conservation when compared to the results obtained using the CR2 scheme. The AVC scheme presented in this paper can be easily implemented within the framework of any numerical approach to solving Eqs. (1) and (2) on non-uniform and even unstructured meshes as well.

The rest of this paper is organized as follows. Section 2 provides the theoretical background and formulation of the AVC scheme in a generalized manner. Section 3 gives details of the specific implementation of the AVC scheme applied within the framework of numerical methods proposed by Nourgaliev and Theofanous [3] used