Journal of Computational Physics 272 (2014) 279–306

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A cell based particle method for modeling dynamic interfaces

Sean Y. Hon [a], Shingyu Leung [a,*], Hongkai Zhao [b]

[a] Department of Mathematics, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
[b] Department of Mathematics, University of California at Irvine, Irvine, CA 92697-3875, United States

ARTICLE INFO

Article history:
Received 22 August 2013
Received in revised form 9 April 2014
Accepted 14 April 2014
Available online 23 April 2014

Keywords:
Interfacial flows
Interface tracking
Eulerian underlying mesh
Lagrangian sampling particles

ABSTRACT

We propose several modifications to the grid based particle method (GBPM) [21] for moving interface modeling. There are several nice features of the proposed algorithm. The new method can significantly improve the distribution of sampling particles on the evolving interface. Unlike the original GBPM where footpoints (sampling points) tend to cluster to each other, the sampling points in the new method tend to be better separated on the interface. Moreover, by replacing the grid-based discretization using the cell-based discretization, we naturally decompose the interface into segments so that we can easily approximate surface integrals. As a possible alternative to the local polynomial least square approximation, we also study a geometric basis for local reconstruction in the resampling step. We will show that such modification can simplify the overall implementations. Numerical examples in two- and three-dimensions will show that the algorithm is computationally efficient and accurate.

1. Introduction

Interfacial motion can be found in many applications in science. It is therefore very important to develop an efficient and flexible yet simple numerical representation for various modelings. Roughly speaking, we can classify existing numerical methods for moving interface problems into two categories according to the way how the interface is numerically represented. The first category is tracking methods, in which the interface is *explicitly* represented by Lagrangian marker particles and its dynamics is tracked by the motion of these particles. For example, boundary integral methods [14,28], boundary element methods, front tracking method [10,37,41], etc. belong to this type. The second class is capturing methods, in which the interface is *implicitly* embedded in a scalar field function defined on a fixed mesh, such as a Cartesian grid. The interface dynamics is captured by the evolution of the scalar function in an Eulerian framework. Main representatives include the level set methods [27], phase field method [1,5], volume of fluid method [13], to name just a few. Recently, there are various interesting works to combine these Lagrangian and Eulerian approaches for moving interface problems, such as the level contour reconstruction methods [32,33], a front tracking method with an underlying grid [9], the particle level set method [6] and some related methods [12], the dynamic surface extension method [36], the fixed grid method in [30] and a closely related vector level set method in [40].

In [21] we have proposed a novel framework to model interface motions. The method naturally combines and takes advantages of both the Lagrangian (explicit) and the Eulerian (implicit) formulations. The basic idea is to represent and track the interface explicitly as in the usual Lagrangian methods using quasi-uniform meshless particles, while an underlying Eulerian grid serves as a reference for those particles. Even though the original GBPM [20–22] has already shown some

* Corresponding author.
E-mail addresses: yshon@ust.hk (S.Y. Hon), masyleung@ust.hk (S. Leung), zhao@math.uci.edu (H. Zhao).

http://dx.doi.org/10.1016/j.jcp.2014.04.032
0021-9991/

promising results, we would like to propose several new modifications to the method to further improve its efficiency and its ease in implementation. According to [21], the GBPM provides an automatic redistribution of the sampling particle according to the underlying mesh. These sampling points (footpoints) are chosen to be the L^2-projection of all grid points in a neighborhood of the interface. We can therefore obtain a quasi-uniform sampling of the interface. Although we can control the maximum distance between two adjacent footpoints on the interface, the non-uniform behavior comes from the fact that there is no mechanism to bound the distance from below, i.e. we do not have any bound like $|\mathbf{y}_i - \mathbf{y}_{i+1}| > O(h)$ where $\mathbf{y}_i$ and $\mathbf{y}_{i+1}$ are two adjacent footpoints, and h is the underlying mesh size. For example, considering a uniform underlying Cartesian mesh in the two dimensional cases with a straight interface parallel to an axis direction, we obtain multiple footpoints projected onto the same locations. This means that $(\mathbf{y}_i - \mathbf{y}_{i+1})$ could in fact be zero and the footpoint-grid point map may be far from one-to-one.

Another challenge to the GBPM is that it is not straight-forward to develop high order accurate numerical quadrature rules along the interface. Surface integrals arise in many circumstances including the boundary integral method [14,28] and the boundary element method [3], where the solution is represented by integrals along its interface. The original GBPM proposed in [21] can indeed evaluate the integral by approximating the interface by piecewise connected linear function. For instance, we can keep track of a global parametrization associated to each footpoint. This provides an ordering of the interface which can be used to approximate Δs_i in the Trapezoidal rule

$$\int_{\Sigma} f(s)ds \simeq \sum_i f_i \Delta s_i.$$

The idea has recently been applied in [22] to integrate Gaussian beam solution to high frequency wave propagation. However, since the interface is approximated using only a low order reconstruction, the integration results are in general less accurate. Another approach has recently been used in [23] by first converting the GBPM representation to a signed distance function as in the level set method [27]. Then the integral can be computed using the δ-function formulation.

One main contribution of this paper is the following modifications to the original GBPM method which can naturally decompose the interface into a summation of high order disconnected segments. Given a computational domain Ω, we partition it into a union of *cells*, i.e. $\Omega = \bigcup_{i,j} \Omega_{i,j}$. If we divide the domain into a sum of rectangles with sides Δx and Δy, we can interpret this *cell*-type discretization as the staggered version of the original discretization described in [21]. The main difference here is that we denote $\mathbf{p}_{i,j}$ the cell-center, rather than the vertices of a cell (corresponding to a grid point in [21]). Moreover, in the case of triangulating Ω into a sum of triangles or tetrahedral, this *cell*-based is more natural. Relating to the *cell*-based domain partition, one simple modification is to activate only those *cells* which contain a segment of the interface. This activation criterion automatically partition the interface into a summation of segments. There are advantages of such an interface partition representation. For example, to visualize the solution in the level set representation, one has to post-process the level set function in order to extract the interface. In the original GBPM, one either simply plots the computed unconnected footpoints or reconstructs a level set representation in order to get an explicit curve for the same interface. In this proposed representation, we have a local closed form representation of the interface within each cell. This allows us to easily approximate any integral within each cell. In this paper, we will further apply the method to study some integral-dependent evolutions of an interface. This application will be important in various fields including multiphase flow modeling using a weak formulation [2], or high frequency asymptotic solutions to the wave equation or the Schrödinger equation [18,19,22]. Since this overall algorithm is a cell-based representation, we name the proposed numerical approach the Cell-Based Particle Method (CBPM).

To simplify the resulting algorithm from the above activation criterion, we can simply replace the L^2-projection in the definition of the footpoint [21] by the L^∞-projection in the Cartesian mesh, or the projection based on the area coordinates in the triangular mesh. If the underlying cells are Cartesian, we can determine the footpoints by

$$\min_s \left\| \mathbf{f}(s) - \mathbf{p} \right\|_\infty,$$

where $\mathbf{p}$ is the coordinates of the mesh point and $\mathbf{f}$ is the parametrization of the interface. If this L^∞-distance is smaller than $\Delta x/2$ (for $\Delta x = \Delta y$), the cell intersects with the interface. This condition therefore naturally matches with the activation condition in practice.

In the original GBPM, the reconstruction is done in a local coordinate system where one assumes the interface can be treated as a graph and is well-approximated by a local polynomial. This can still be easily done in the current CBPM representation. In this work, we also study an alternative to this *extrinsic* polynomial least squares fitting by considering the *intrinsic* fitting based on geometric basis. In particular, we locally approximate the interface in two dimensions using straight lines or circles. Therefore, no local coordinate system is required. Furthermore, since the interface is approximated using such simple geometric basis, it significantly simplifies many numerical differential operators defined on the manifold.

The paper is organized as follows. We will briefly summarize the GBPM in Section 2. In Section 3 we will introduce the proposed CBPM. Detailed descriptions of various steps in the algorithm will also be given. To further simplify various part of the algorithm, we propose in Section 4 to incorporate the geometric basis for local reconstruction. Convergence tests and numerical examples are given in Section 5 to demonstrate the effectiveness of the method.