Journal of Computational Physics 300 (2015) 365–386

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Accurate interface normal and curvature estimates on three-dimensional unstructured non-convex polyhedral meshes

Christopher B. Ivey *, Parviz Moin

Center for Turbulence Research, Stanford University, Stanford, CA 94305, USA

ARTICLE INFO

Article history:
Received 27 January 2015
Received in revised form 13 July 2015
Accepted 17 July 2015
Available online 31 July 2015

Keywords:
Two-phase flow
Volume of fluid
Height function
Unstructured grid
Curvature
Non-convex polyhedron

ABSTRACT

This paper presents a framework for extending the height-function technique for the calculation of interface normals and curvatures to unstructured non-convex polyhedral meshes with application to the piecewise-linear interface calculation volume-of-fluid method. The methodology is developed with reference to a collocated node-based finite-volume two-phase flow solver that utilizes the median-dual mesh, requiring a set of data structures and algorithms for non-convex polyhedral operations: truncation of a polyhedron by a plane, intersection of two polyhedra, joining of two convex polyhedra, volume enforcement of a polyhedron by a plane, and volume fraction initialization by a signed-distance function. By leveraging these geometric tools, a geometric interpolation strategy for embedding structured height-function stencils in unstructured meshes is developed. The embedded height-function technique is tested on surfaces with known interface normals and curvatures, namely cylinder, sphere, and ellipsoid. Tests are performed on the median duals of a uniform cartesian mesh, a wedge mesh, and a tetrahedral mesh, and comparisons are made with conventional methods. Across the tests, the embedded height-function technique outperforms contemporary methods and its accuracy approaches the accuracy that the traditional height-function technique exemplifies on uniform cartesian meshes.

1. Introduction

The volume-of-fluid (VOF) method is one of the most widely used formulations to simulate interfacial and free-surface flows [1]. In this method, the interface evolution is implicitly tracked using a discrete function, F, representing the volume fraction of the tagged fluid within a cell of the computational mesh. F is a discretized version of the fluid marker function, f, that is constant in each phase, jumps at the interface from 0 to 1, and follows the scalar advection equation,

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

where $\vec{v}$ is the velocity vector.

* Corresponding author. Tel.: +1 757 570 2369.
E-mail addresses: civey@stanford.edu (C.B. Ivey), moin@stanford.edu (P. Moin).

http://dx.doi.org/10.1016/j.jcp.2015.07.055
0021-9991/

The piecewise-linear interface calculation (PLIC) has become the standard interface representation within the VOF community [2]. PLIC-VOF methods describe the interface by a series of disconnected planes, each oriented by a unit normal, $\hat{n}$, and positioned by a constant, C, such that $\hat{n} \cdot \vec{x} + C = 0$. Two key steps in any PLIC representation include volume truncation by a fixed plane, i.e. determination of F given $\hat{n}$ and C, and volume enforcement by a movable plane, i.e. determination of C given $\hat{n}$ and F. The importance of the volume truncation and enforcement operations has led researchers to develop analytic and geometric tools to expedite computations for rectangular and hexahedral elements [3], for triangular and tetrahedral elements [4], and for convex polyhedral elements [5]. In this paper, we extend the class of geometric tools to non-convex polyhedral meshes in order to implement the PLIC-VOF method in a collocated node-based finite-volume flow solver [6]. As evidenced by the volume enforcement and truncation operations, estimation of $\hat{n}$ is key to the accuracy of any PLIC-VOF method.

The equations governing the motion of an unsteady, viscous, incompressible, immiscible two-fluid system are the Navier–Stokes equations, augmented by a localized surface tension force,

$$\rho\left(\frac{\partial \vec{v}}{\partial t} + \vec{v}\cdot\nabla\vec{v}\right) = -\nabla p + \nabla\cdot\left(\mu\left[\nabla\vec{v} + \left\{\nabla\vec{v}\right\}^T\right]\right) - \sigma\kappa\hat{n}\delta_S,$$
$$\nabla\cdot\vec{v} = 0,$$
$$\kappa = \nabla\cdot\hat{n}, \tag{2}$$

where, ρ is the density, p is the pressure, μ is the viscosity, σ is the surface tension coefficient, κ is the interface curvature, and δ_S is the Dirac Delta function localized to the surface S. As evidenced by Eq. (2), in addition to accurately estimating $\hat{n}$, the PLIC-VOF framework also needs to calculate the rate at which $\hat{n}$ turns along the interface, i.e. the curvature, κ.

Determination of $\hat{n}$ and κ in the VOF method is problematic due to the discontinuous nature of F. Nevertheless, various algorithms to calculate $\hat{n}$ and κ have been proposed. The traditional Parker–Youngs (PY) method [7] uses simple difference formulas to calculate gradients in F for the estimation of $\hat{n}$. The method has been implemented on nonorthogonal [8] and unstructured meshes [9]; however, the PY method is at most first-order accurate because $\hat{n}$ for a rectilinear interface is not calculated exactly, a necessary condition for second-order accuracy [10]. The least-squares fit procedure [11,12] is more accurate than the PY method and has been extended to unstructured meshes [13]; however, it too does not satisfy the necessary condition for a second-order method. Several second-order methods for estimating $\hat{n}$ have been proposed, namely the least-squares volume-of-fluid interface reconstruction algorithm (LVIRA) and the efficient least-squares volume-of-fluid interface reconstruction algorithm (ELVIRA) for structured grids [10], and the geometric least-squares (GLS) method for unstructured grids [14], each able to reconstruct a rectilinear interface exactly. LVIRA orients $\hat{n}$ such that the discrepancy in F from using the calculated linear interface over a neighborhood of cells is minimized in the least-squares sense. The procedure requires costly geometric iterations in which volume enforcement and volume truncation steps must be performed for each cell. ELVIRA bypasses the iterations by selecting $\hat{n}$ amongst a set of candidates constructed from the centered, backward, and forward estimates in each direction. The GLS method follows the procedure of LVIRA, requiring geometric iterations within an unstructured framework – a prohibitively costly procedure. A well-known non-iterative method for estimating $\hat{n}$ is the height-function (HF) technique. In the HF method, F is integrated in the cartesian direction closest to $\hat{n}$ (approximated with a simpler method) to calculate a height, H. Slopes of a local H distribution in the other two cartesian directions are used to correct $\hat{n}$ [15–17]. The HF method was shown to be second-order accurate with proper handling for particular alignments of the interface with respect to the grid lines [18,19]. In two dimensions, the method was extended to nonuniform rectangular grids [20] and, by adapting the definition of H and using a least-squares fit, to unstructured rectangular/triangular grids [21], both exhibiting second-order convergence in $\hat{n}$.

As shown in Eq. (2), κ requires higher differentiability than that of $\hat{n}$. To address the lack of differentiability of F, various methods have been posited to calculate κ. In the convolved VOF (CV) method, F is convolved with a smoothing kernel to provide a smoothed-out distribution from which the second derivates can be calculated [15,22]. The reconstructed-distance function (RDF) method builds a signed-distance function away from the interface to provide a smooth field from which κ can be calculated [15]. The RDF technique was extended to unstructured rectangular/triangular meshes [23]. Both the CV and RDF methods have shown lack of convergence under refinement on structured [15] and unstructured [21] meshes. In addition to the calculation of $\hat{n}$, the HF method has been used to calculate κ [15], demonstrating second-order accuracy over a series of canonical test problems on uniform cartesian meshes [18,24,25]. In two dimensions, the calculation of κ with the HF method was extended to nonuniform rectangular grids without loss of the second-order convergence [20]. The HF technique was extended to two-dimensional unstructured rectangular/triangular grids [21]; however, the reframed definition of H required quadric fitting to calculate κ, and the method was less than first-order accurate. An HF method with $\hat{n}$-aligned columns of variable H was shown to improve the κ calculation accuracy on coarse meshes; however, it required a neighborhood search step to compute intersections with the mesh and the columns [26]. The HF method was also adapted to quad and octree discretizations [27], where cartesian stencils of varying H (in addition to parabolic fitting for stencils with ill-defined H) were used to obtain second-order accuracy. To the best of our knowledge, the HF technique for calculating $\hat{n}$ and κ has not been extended to three-dimensional unstructured meshes. Furthermore, a convergent method to calculate κ on unstructured meshes has not been published.

In this paper, we extend the HF technique for estimating $\hat{n}$ and κ to three-dimensional unstructured non-convex polyhedral meshes. The method embeds structured HF stencils in the unstructured mesh and interpolates the mesh F data to