Journal of Computational Physics 257 (2014) 645–669

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A balanced-force algorithm for two-phase flows

Hanif Montazeri *, C.A. Ward

Thermodynamics and Kinetics Laboratory, Department of Mechanical & Industrial Engineering, University of Toronto, Canada

ARTICLE INFO

Article history:
Received 25 July 2013
Accepted 30 September 2013
Available online 17 October 2013

Keywords:
Multiphase flow
Body forces
Non-uniform grids
Pressure–velocity coupling
Fifth-order WENO method
Marangoni stress

ABSTRACT

Numerical methods for imposing body forces in two-phase flow simulations are discussed. Numerical schemes are presented to avoid the inaccurate solutions that result from inconsistent implementation of forces. First, the momentum equations are discretized so that they accurately accommodate the discontinuity in fluid properties at an interface. Consistent numerical estimations for different body forces such as interfacial (including Marangoni), gravity and electromagnetic forces are discussed. Then, it is shown that the standard pressure–velocity coupling scheme for collocated algorithms is not sufficient for multiphase flows, and therefore a new pressure–velocity coupling is devised and tested for both single and two-phase flows. Finally, to advect the level set function, a cost effective fifth-order WENO method is developed. These formulations are accurate and efficient both for uniform and non-uniform meshes. Several test cases are presented and compared with a standard implementation of body forces to demonstrate the efficiency of the proposed algorithm.

Crown Copyright © 2013 Published by Elsevier Inc. All rights reserved.

1. Introduction

In the numerical simulation of two-phase flows, there are two types of body forces: one is the replacement of the dynamic boundary conditions between the two phases, such as surface tension, and the other is the volumetric body forces, such as gravity or electromagnetic forces. The first type, known as interfacial forces, is singular and therefore poses a great challenge in a numerical simulation. Volumetric body forces can also challenge numerical schemes because of the discontinuity of material properties such as density and viscosity in two-phase flows. Two-phase electrohydrodynamic flows, for instance, are involved with implementing discontinuous electric forces. Nonetheless, forces caused by a uniform potential field (e.g. gravity) even acts as a discontinuous body force (because of the density jump at the interface). Tomar et al. [37] and Montazeri et al. [26] introduced numerical methods to minimize the errors produced by these non-interfacial forces. They discussed the importance of devising numerical schemes to eliminate the numerical errors which can deteriorate flow simulations.

One of the first successful methods in implementing interfacial forces was the Ghost Fluid Method (GFM) [19,22]. GFM uses a sharp treatment of the interface and therefore utilizes singular forces to capture pressure jump. Perhaps the second most successful method was the balanced-force algorithm introduced by Francois et al. [12]. They introduced a straightforward algorithm for both continuum and sharp treatment of the two-phase flow discontinuities. Their sharp method, however, was inherently GFM. Based on the formulation presented in [12], the similarity of numerical operators used to calculate the pressure gradients and the interfacial forces seemed to be the key for the successful results [15]. Montazeri et al. [26] extended the balanced force algorithm for both interfacial and non-interfacial forces and showed that the similarity of numerical operators is not a necessary condition for successful results (similar conclusion is also found in this paper).

* Corresponding author.
E-mail address: hanif@mie.utoronto.ca (H. Montazeri).

0021-9991/$ – see front matter Crown Copyright © 2013 Published by Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2013.09.054

However, both algorithms are derived for only equidistance computational grids and fail to produce successful results for non-equidistance grids. Using a uniform mesh, however, is not practical in many engineering and scientific applications.

Utilizing non-uniform [11] or adaptive mesh [44] are two common methods for reducing computational cost of numerical simulations. Nonetheless, accurate and efficient adaptive mesh techniques are developed based on using non-uniform grids. These non-uniform adaptive mesh techniques [42,43,45] provide great flexibility in numerical simulations. The first step toward developing such an efficient algorithm for two-phase flows is to develop accurate discretization on static non-uniform grids. Using static non-uniform grids is a common method for reducing computational cost in many scientific codes. Therefore, the goal of this paper is to present an accurate and cost effective algorithm for implementing different body forces in a two-phase flow when non-equidistance grids are used. This, as will be shown, needs substantial modifications in different parts of flow algorithm. The main parts of such modifications are: consistent implementation of body forces, the design of new pressure–velocity coupling and the development of cost effective interface capturing technique.

It is well-known that when flow equations are solved on a collocated mesh, a spurious pressure mode can appear unless modifications are used to suppress the erroneous modes [11]. Over the years, different numerical methods have been proposed to avoid the undesirable oscillations in the pressure field [1,4–10,20,24,29,31,33,35]. The best known of these was introduced by Rhie and Chow [34]. Mimicking the staggered formulation, they used a momentum-based interpolation for the face velocities in the continuity equation of the SIMPLE algorithm. Using a Taylor series expansion, Rahman et al. [30] rigorously derived and modified the Rhie and Chow [34] interpolation, to account for the influence of a non-pressure gradient source terms. This modification can be important for flows with strongly varying body forces, such as in buoyant and swirling flows. Within the fractional-step method [21], Tafti [36] developed and tested different finite-difference formulations to discretize Laplacians to reduce pressure oscillations and at the same time satisfy discrete continuity. It was found that a third-order forward-biased approximation for the gradient operator of pressure can effectively eliminate pressure oscillations in a lid-driven cavity test case [36]. A similar treatment was proposed by Rauwoens et al. [32] to include density variations in a flow field. They present a clear mathematical derivation to obtain an effective pressure–velocity coupling in the framework of pressure–correction algorithms. Finally, the method for properly closing equations (MPCE) of Ashrafizadeh et al. [2] is a systematic co-located finite-volume discretization scheme that employs a single interpolation formula for both convected and convecting velocity components.

Most of these pressure–velocity coupling methods can be extended to non-uniform meshes; however they might not be the most efficient methods. So far as we know, the Rhie and Chow [34] method is the only one that has been extensively used for non-uniform meshes [11]. The inexpensive, straightforward, non-uniform formulation of this scheme has made it a suitable technique for many discretizations. However, as will be detailed in Section 4, one of the main assumptions in this formulation is the linearity of pressure profile. This is not a valid assumption in the context of two-phase flows. Nevertheless, the method even fails to reach the exact solution for a linear pressure profile (see Section 4).

In addition to pressure–velocity coupling schemes, efficient discretization of interface capturing techniques on non-uniform grids needs to be devised. In this paper, we use the Level set function to capture the interface position of a two-phase flow. Numerical experiments show that the numerical solution of the level set equations is very sensitive to the way the spatial terms are discretized [27]. Therefore highly accurate numerical schemes are popular for discretizing the level set equation. Successful Essentially Non-Oscillatory (ENO) [14] schemes are the most common type of discretization seen in the literature. From the same family, Weighted ENO schemes (WENO) were first introduced by Liu et al. [23] and then modified, and improved to a 5th-order accurate scheme by Jiang and Shu [18]. Jiang and Peng used WENO schemes to approximate the viscosity solution of the Hamilton–Jacobi equation [17]. Further developments of higher-order schemes are presented in [13] and [41]. Also, Wolf and Azevedo [39] described the implementation and analysis of ENO and WENO schemes for unstructured grids. Wang et al. [38] extended the fifth-order WENO method to non-uniform meshes. They developed explicit formulas for a fifth-order WENO method within the framework of a finite-volume technique. However, this discretization is very expensive [37], and therefore in this paper we develop an alternative finite-difference method to implement the level set function.

In Section 2, we briefly review the governing equations of two-phase flows. The implementation of generic body forces (e.g. Marangoni, electromagnetic and gravity forces) is described in Sections 3 and 4. An alternative pressure–velocity coupling technique method is presented in Section 5, and a new implementation of the level set function is detailed in Section 6. We conclude with the results of several test cases to demonstrate the robustness and accuracy of the numerical schemes developed in this paper.

2. Governing equations

To derive governing equations for two-phase flows, we start with the basic equations for conservation of mass and momentum for a single fluid with non-constant properties:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho U_i) = 0 \tag{2.1}$$

$$\frac{\partial}{\partial t}(\rho U_i) + \frac{\partial}{\partial x_j}(\rho U_j U_i) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\right] + \rho g_i + \sum \mathcal{T}_i(\rho, \xi, \ldots) \tag{2.2}$$

Download English Version:

https://daneshyari.com/en/article/10356010

Download Persian Version:

https://daneshyari.com/article/10356010

Daneshyari.com