Journal of Computational Physics 282 (2015) 269–288

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Parallel load balancing strategy for Volume-of-Fluid methods on 3-D unstructured meshes

Lluís Jofre [a], Ricard Borrell [a,b], Oriol Lehmkuhl [a,b], Assensi Oliva [a,*]

[a] Heat and Mass Transfer Technological Center (CTTC), Universitat Politècnica de Catalunya – Barcelona Tech, ETSEIAT, Colom 11, 08222 Terrassa, Barcelona, Spain
[b] Termo Fluids S.L., Av. Jacquard 97 1-E, 08222 Terrassa, Barcelona, Spain

ARTICLE INFO

Article history:
Received 10 December 2013
Received in revised form 20 October 2014
Accepted 1 November 2014
Available online 13 November 2014

Keywords:
Interface-capturing
Load balancing
Parallelization
Unstructured mesh
Volume-of-Fluid

ABSTRACT

Volume-of-Fluid (VOF) is one of the methods of choice to reproduce the interface motion in the simulation of multi-fluid flows. One of its main strengths is its accuracy in capturing sharp interface geometries, although requiring for it a number of geometric calculations. Under these circumstances, achieving parallel performance on current supercomputers is a must. The main obstacle for the parallelization is that the computing costs are concentrated only in the discrete elements that lie on the interface between fluids. Consequently, if the interface is not homogeneously distributed throughout the domain, standard domain decomposition (DD) strategies lead to imbalanced workload distributions. In this paper, we present a new parallelization strategy for general unstructured VOF solvers, based on a dynamic load balancing process complementary to the underlying DD. Its parallel efficiency has been analyzed and compared to the DD one using up to 1024 CPU-cores on an Intel SandyBridge based supercomputer. The results obtained on the solution of several artificially generated test cases show a speedup of up to $\sim 12\times$ with respect to the standard DD, depending on the interface size, the initial distribution and the number of parallel processes engaged. Moreover, the new parallelization strategy presented is of general purpose, therefore, it could be used to parallelize any VOF solver without requiring changes on the coupled flow solver. Finally, note that although designed for the VOF method, our approach could be easily adapted to other interface-capturing methods, such as the Level-Set, which may present similar workload imbalances.

1. Introduction

The numerical simulation of immiscible multi-fluid flows is fundamental to a better understanding of many phenomena of interest in different disciplines such as engineering, hydrodynamics, geophysics or fundamental physics. Typical examples are the simulation of sprays, injection processes, bubbles, breakup of drops, wave motion, etc. These type of flows are characterized by the existence of an interface, separating the different fluids, which needs to be reproduced by the simulation method. So far, different numerical methods exist to reproduce the interface motion. These can be classified into two main groups: interface-tracking and interface-capturing methods. On the one hand, the interface-tracking approaches chase the interface as it moves: (1) defining the interface as a boundary between subdomains of a moving mesh [1–3]; (2) following

* Corresponding author. Tel.: +34 93 739 81 92; fax: +34 93 739 89 20.
E-mail address: cttc@cttc.upc.edu (A. Oliva).

http://dx.doi.org/10.1016/j.jcp.2014.11.009
0021-9991/

the Lagrangian trajectories of massless particles [4–6]. On the other hand, the interface-capturing approaches describe the motion of the interface by embedding the different fluids into a static mesh. In particular, from this last group, the two main options of choice are the Volume-of-Fluid (VOF) [7–9] and Level-Set (LS) [10–12] methods, as well as algorithms based on combinations of both. From all these options, this paper is focused on the VOF method. This is based on geometrically reconstructing the fluids interface and, according to it, evaluating the portion of advected volumetric flux corresponding to each fluid. Its major strength is the accuracy achieved by some of its implementations on capturing sharp interfaces and their complex deformation, including breakups, while complying with the volume preservation constraint. This accuracy results in high computational costs. However, in the last decade, with the increase of the available computing power, different interfacial problems have been successfully tackled using it. Examples are the simulation of the drop breakup phenomenon by Renardy [13], the bubble motion by Annaland et al. [14], the solution of wave impact problems by Kleefsman et al. [15] or the numerical study of primary and impinging jet atomizations by Fuster et al. [16], Tomar et al. [17] and Chen et al. [18].

In general, on the simulation of interfacial multi-fluid flows with VOF methods, the computing costs are dominated by the Navier–Stokes (NS) flow solver, and specifically by the solution of the Poisson system derived from the incompressibility constraint. Even so, the cost of the VOF calculations is not negligible at all. Its relative weight depends on different factors, such as the algorithms chosen, the effectiveness of its implementation, the physical case being considered, the type of geometric discretization used, the computing system employed, etc. As an example, on the sequential simulation of the Richtmyer–Meshkov instability [19] with an unstructured tetrahedral mesh of 250K cells, our VOF solver represents 22% of the computing costs. A similar percentage was reported by Le Chenadec and Pitsch [20], on the solution of a diesel jet with a Cartesian mesh of $256 \times 256 \times 1152$ cells. Anyway, beyond the percentage obtained for any particular simulation, it is a certainty that, in the high performance computing context, the cost of the VOF calculations will become more and more important while the algorithmic solutions adopted disregard parallel performance issues. Besides, by contrast, many efforts are employed by the scientific community on the parallelization of NS flow solvers and, in particular, on Poisson solvers [21,22]. Considering, for example, the aforementioned Richtmyer–Meshkov instability case, with the DD approach we measured a raise of the VOF cost up to 84% when engaging 128 CPU-cores while, with the new parallelization strategy presented in this paper, the percentage is kept at 24%.

The parallel performance limitations of the standard DD approach can also be observed in the work by Aráujo et al. [23] focused on the 3-D simulation of injection processes. Their tests show a maximum parallel efficiency of 50% with up to 80 CPU-cores, including both the momentum and the VOF solvers. Another study on parallel algorithms for multiphase flows is the work of Sussman [24], based on solving the pressure Poisson equation by means of a multi-level solver and the interface motion through a coupled LS and VOF method [25]. This last work, however, is mainly focused on the performance of the pressure solver and, after all, no more than 16 CPU-cores were used in the parallel performance tests. Surprisingly, we could not find other relevant works on the literature presenting new alternatives for the parallelization of VOF methods.

Broadening the literature search to LS-based interface-capturing approaches, we found an additional parallelization alternative studied by Herrmann [26], which may be adapted to VOF methods. In particular, LS methods require the solution of an extra partial differential equation (PDE) to maintain the interface sharp. Similarly to VOF methods, this interface re-initialization process is not well balanced if the interface is not homogeneously distributed throughout the domain. Herrmann proposes to generate two independently adapted grids for the solution of the flow and interface motion, respectively. While no restrictions are imposed on the Navier–Stokes grid, an equidistant Cartesian grid is adopted for the interface motion solution, with enough resolution to ensure accuracy of the LS method at any part of the domain, avoiding the application of complex adaptive mesh refinement (AMR) algorithms. This configuration also simplifies the LS parallelization since, in order to achieve a good workload balance, tasks can be easily reassigned between parallel processes without geometric information exchange. This approach was tested on the solution of Zalesak's disk case, obtaining a slightly sub-optimal speedup with up to 128 CPU-cores [26]. In a later work, Herrmann applied the same strategy on a multi-scale Eulerian/Lagrangian two-phase flow algorithm [27], where the LS grid method was used for the Eulerian part, the overall algorithm showed an excellent speedup with up to 2048 CPU-cores.

Therefore, considering the good parallel performance achieved by Herrmann with his load balancing strategy, our purpose has been to develop a similar strategy for the parallelization of VOF methods on general unstructured meshes. Moreover, we solve both the motion of the flow and of the interface in the same mesh, without imposing any restriction to it. Consequently, the load balancing algorithm and its computing profile undergo major changes with respect to the Herrmann approach. For instance, when a task is reassigned, in the Cartesian case no geometric information needs to be transmitted since the mesh is homogeneous, contrary, in the general case the geometric characteristics of the discrete elements engaged on the task need to be transmitted as well. Additionally, our load balancing approach is based on a precise optimization algorithm, rather than iteratively reassign tasks until some threshold imbalance is reached or the process stalls. Finally, note that although our algorithm has been developed for VOF methods, it could be easily adapted to the parallelization of LS methods on unstructured grids.

Hence, this paper presents a new strategy for the parallelization of VOF methods on unstructured meshes, which is based on a dynamic load balancing process complementary to the DD. The rest of the document is organized as follows: in the next section, the mathematical formulation of the VOF method on unstructured meshes is presented. The standard domain decomposition and our new load balancing parallelization strategy are detailed in Section 3. An exhaustive analysis and comparison of the parallel performance issues of both methods are presented in Section 4. Finally, conclusions are drawn in Section 5.