AN OPTIMIZED SCALABLE PARALLEL FRONT-TRACKING METHOD FOR SIMULATION OF MULTIPHASE FLOWS

Z. Ahmed[*], D. Izbassarov[†], M. N. Farooqi[‡], D. Unat[‡], M. Muradoglu[*]

[*] Koc University, Department of Mechanical Engineering
34450 Istanbul, TURKEY. E-mail: mmuradoglu@ku.edu.tr

[†]Royal Institute of Technology (KTH), Department of Mechanics
SE-100 44, Stockholm, SWEDEN.

[‡]Koc University, Department of Computer Engineering
34450 Istanbul, TURKEY.

Key words: Front-tracking method, multiphase flow, scalability, optimization

Abstract. We present a scalable parallelization of a front-tracking method and its application to study effects of soluble surfactant on the lateral migration of deformable bubbles in a pressure-driven channel flow. The front-tracking method uses an Eulerian grid to solve the flow equations and a Lagrangian grid to track the interface, so it poses a challenging task to develop a scalable parallelization algorithm due to different types of communication within and between the Eulerian and Lagrangian grids. In this study, we develop a parallel algorithm and propose two different strategies for handling communication between the Lagrangian and Eulerian grids. Strong and weak scaling studies are performed and good scalability is demonstrated on two supercomputers. Then extensive simulations are performed to examine the effects of soluble surfactant on the lateral migration of deformable bubbles in a pressure-driven channel flow.

1 Introduction

Multiphase flows are ubiquitous in natural processes and industrial applications. It is therefore crucially important to develop computational tools for accurate simulations of multiphase flows in realistic flow conditions. The main difficulty arises from the existence of interface separating different phases that evolves continuously with the flow and may undergo large deformations often leading to topological changes such as breakup and coalescence. Tracking of fluid interfaces has proved to be a notoriously difficult task. In the front-tracking method [8], the interface is explicitly tracked using a Lagrangian grid consisting of marker points moving with the local flow velocity interpolated from the underlying Eulerian grid on which the flow equations are solved. For accurate simulations, the front-tracking method requires a high spatial and temporal resolution, which in turn demands high computational power. In addition, multiphase flow simulations must be performed using millions of bubbles to acquire meaningful statistical information about the most of practical bubbly flows. Non-uniform distribution of bubble sizes and the multi-physics effects such as soluble surfactant, viscoelasticity, thermocapilarity, etc. add further complexity to the multiphase flow problems. To keep computational time within practical limits simulations must be done in parallel.

In the present study, an efficient parallel algorithm is developed for the front-tracking method. Two different communication strategies are proposed for the data exchange between the Eulerian and Lagrangian grids. Various latency-hiding optimizations are implemented to reduce the communication overhead. Both strong and weak scalabilities are examined on two different hardware systems and the method is demonstrated to be highly scalable. Finally, method is used to study the effects of soluble surfactant on lateral migration of deformable bubbles in pressure-driven channel flow. It is found that the surfactant collected at the interface create the Marangoni stresses that counteract the viscous shear stress and even a small amount of surfactant can alter the direction of the lateral migration of the bubbles and thus the overall behavior of the bubbly flow.

2 Governing Equations and Numerical Method

The front-tracking is based on one-field formulation of the flow equations. In this framework, the incompressible flow equations are written as

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0, & (1)\\ \frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) &= -\nabla p + \nabla \cdot \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T) + \int_A \sigma(\Gamma)\kappa \mathbf{n}\delta(\mathbf{x} - \mathbf{x}_f)dA, & (2)\end{aligned}$$

where $\mathbf{u}$, p, ρ and μ are velocity vector, pressure, density and viscosity fields, respectively. The surface tension is treated as a body force in the last term on the right hand side of Eq. (2), where σ is the surface tension coefficient determined as a function of the interfacial surfactant concentration Γ, κ is twice the mean curvature, and $\mathbf{n}$ is a unit vector normal to the interface. The three-dimensional delta function δ indicates that the surface tension acts only on the interface and its arguments $\mathbf{x}$ and $\mathbf{x_f}$ denote the point at which the equation is evaluated and a point at the interface, respectively.

In the front-tracking method, the interface is tracked explicitly using a Lagrangian grid while the flow equations are solved on a staggered Eulerian grid as shown in Fig. 1. The Lagrangian grid consists of Lagrangian marker points connected by triangular elements as shown in Fig. 1b. The marker points move with the local flow velocity interpolated from the Eulerian grid. The surface tension is computed on the Lagrangian grid and is then distributed onto the Eulerian grid to be added to the momentum equations as a body force. The indicator function is defined such that it is unity inside the bubble and zero outside, and is used to set the material properties in different phase. The indictor function is computed based on the location of the marker points using the standard procedure as described by Unverdi and Tryggvason [8]. As the interface evolves, it is needed to restructure the Lagrangian grid by deleting the elements that are smaller than a pre-specified lower limit and by splitting the elements that are larger than a pre-specified upper limit in the same way as described by Tryggvason et al. [7]. In the case of contamination,

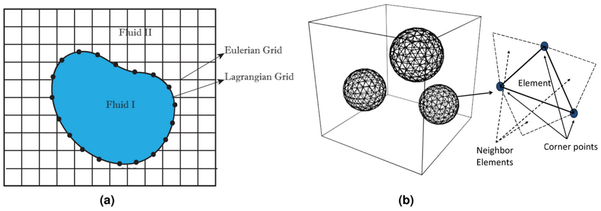


Figure 1: Lagrangian and Eulerian grids (a) in 2D and (b) in 3D. Flow equations are solved on Eulerian grid while interface is represented by a Lagrangian grid consisting of connected marker points [2].

the interfacial surfactant evolution equation is also solved on the Lagrangian grid as described by Muradoglu and Tryggvason [5].

The mass and momentum conservation equations are solved on a staggered Eulerian grid using a projection method [1]. To account for the effects of the soluble surfactant, the bulk and interfacial surfactant concentration evolution equations are solved fully coupled with the flow equations as described by Muradoglu and Tryggvason [5]. Standard second order central finite-difference approximations are used to discretize all the spatial derivatives except for the convective terms for which a third order QUICK scheme is used. The bulk surfactant concentration evolution equation is also discretized and solved on the Eulerian grid. For this purpose, the spatial derivatives are discretized using second order finite-differences except for the convective terms for which a fifth order WENO-Z scheme is employed [5]. Time integration is performed using a second order predictor-corrector scheme. Detailed description of the numerical method can be found in Tryggvason et al. [8] and Muradoglu and Tryggvason [5].

3 Parallelization of the front-tracking method

In parallelization, a data dependency graph is first constructed for the full front-tracking method and then a parallel algorithm is developed focusing on the parallelization of the Lagrangian grid since the Eulerian grid is simply parallelized using a classical domain-decomposition method.

The processes computing the Eulerian and Lagrangian grids are called *Domain* and *Front*, respectively. A simple domain decomposition method is easily applied for the parallelization of the Eulerian grid. Similar to the Eulerian grid, the Lagrangian grids are divided into subgrids, and distribute bubbles among the parallel *Fronts*. The *Front* processor becomes owner of a bubble when the center of the bubble is in the subdomain of the *Front*. Each *Front* is mapped to a number of *Domain* processes and the *Front* communicates only with these *Domains*. An example mapping of 8 *Domains* to 2 *Fronts* is shown in Fig. 2a. As flow evolves, a bubble moves in the physical domain and its center may lie at the border shared by more than one *Front* (Fig. 2b). Obviously shared bubbles complicate parallelization. To deal with this problem, we take the following two simpler approaches: (i) *Owner-computes* in which the shared bubble is computed by the owner *Front*. The owner processor solves the equations, keeps track of the sharers, and sends or receives data while the sharers only route the shared data to the corresponding *Domains*. (ii) *Redundantly-All-Compute* in which *Fronts* containing the shared portion of the bubble redundantly perform computations for the entire shared bubble.

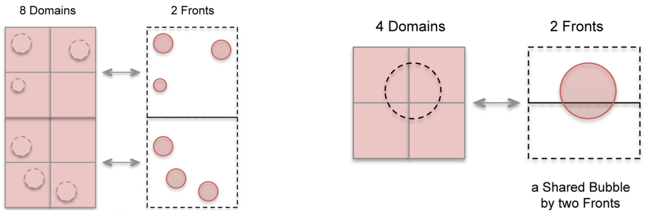
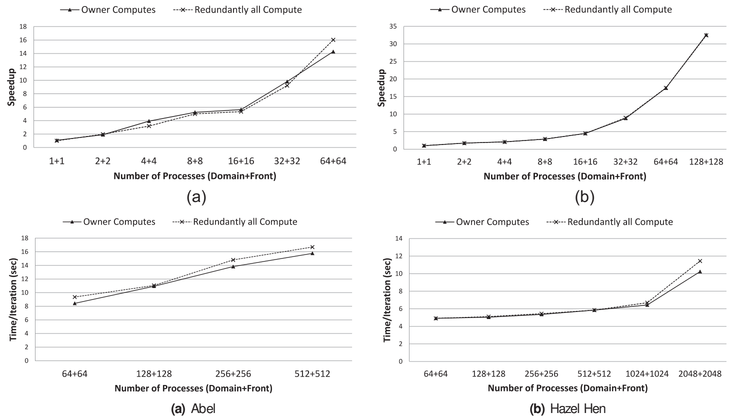


Figure 2: (Left) Work division for parallel *Fronts* where upper 4 *Domains* are assigned to upper *Front* and lower 4 *Domains* are assigned to lower *Front.* (Right) Two parallel *Fronts* with a shared bubble [2].



(a)

(b)

(a) Abel

(b) Hazel Hen

Figure 3: Top row: Strong scaling speedup (higher is better). Bottom row: Weak scaling (lower is better). (Left panel) Abel. (Right panel) Hazel Hen. [2].

4 Results and discussion

An extensive study are carried out to assess the performance of the parallelization algorithm on two supercomputers: The Abel cluster located at the University of Oslo and the Hazel Hen located at the High Performance Computing Centre in Stuttgart. Specifications of both clusters can be found in Farooqi et al. [2]. Simulations are performed for a laminar mono-dispersed bubbly flow in a vertical channel and flow parameters are mainly based on the deformable bubble case studied by Lu and Tryggvason [4]. A reasonable grid resolution is used in which the bubbles are resolved by about 28 × 28 × 28 grid points. Simulations are performed in a cubic periodic box as in Lu and Tryggvason [4].

We first examine the strong scaling and the results are shown in top row of Fig. 3. As seen, for a grid resolution of 256^3, the best speedup for both strategies is achieved on Abel when 64 *Domain*+64 *Front* processes are used. Compared to the baseline case of $1 + 1$ processes, the speedup is 16× for *redundantly-all-compute* and 14× for the *owner-computes strategy.* This figure also shows that the scalability is much better on Hazel Hen mainly due to its higher network bandwidth, i.e., 11.7 GB/s, about twice that of Abel. We next examine the weak scaling of the parallel algorithm. For this purpose, we increase the domain size and number of bubbles while fixing the amount of computational work assigned to each process. The results are shown in bottom row of Fig. 3.

After testing the parallel algorithm, we finally performed extensive simulations to study the effects of soluble surfactant on the lateral migration of the bubble in a pressure-driven channel flow. The computational setup is sketched in the left panel of Fig.4. The

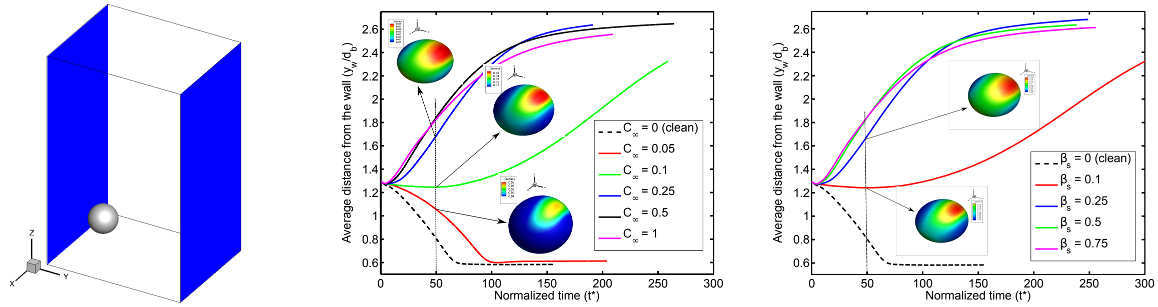


Figure 4: Computational setup for the lateral migration of da eformable bubble in a pressure-driven channel flow (left). Effects of bulk surfactant concentration (middle) and the elasticity number (right).

bubble is initially located near the wall in a pressure-driven fully developed laminar flow. No slip boundary conditions are applied on the walls in the y direction while periodic boundary conditions are applied in all other directions. The bubble is initially clean and the surfactant concentration is uniform in the ambient fluid. Here sample results are presented only for a single bubble. The results for the cases involving many bubbles will be included in the final version of the paper. In many bubble case, topology changes such as bubble coalescence and breakup play an important role in the overall dynamics of bubbly flows and will also be included in the final version of the paper. Figure 4 shows the effects of the bulk surfactant concentration (middle panel) and the elasticity numbers (right panel), respectively. As seen, the bubble moves toward the wall when the bubble is clean due to Magnus effect. In the case of contamination, Marangoni stresses develop at the bubble interface and push the bubble toward the channel centerline. When the Marangoni stresses overcome the aerodynamic lift force, the direction of the lateral migration of the bubble changes completely. These results show that even a small amount of surfactant may be sufficient to alter the direction of the lateral migration of the bubble, which might be of crucial importance in engineering applications including the heat exchangers where it is important to keep the surface of the heat exchanger clean of the bubbles to avoid insulating effects of the bubble layer.

REFERENCES

[1] Chorin A.R. *Mathematics of Computation* (1968) **22**:745–762.

[2] Farooqi et al. *Int. J. High Perform. Comput. Appl.* (2019) **199**: 24-4-5

[3] Lu et al. *Int. J. Multiphase Flow* (2017) **95**:135–143.

[4] Lu, J. and Tryggvason, G. *Phys. Fluids* (2008) **20**:(040701-1)–(040701-6).

[5] Muradoglu, M. and Tryggvason, G. *J. Comput. Phys.* (2014) **274**:737–757.

[6] Muradoglu, M. and Tryggvason, G. *J. Comput. Phys.* (2008) **227(4)**:2238–2262.

[7] Tryggvason et al. *J. Comput. Phys.* (1992) **169(2)**:708–759.

[8] Unverdi, S. and Tryggvason, G. *J. Comput. Phys.* (1992) **100(1)**:25–37.