A Fixed-Mesh Approach for Gas-Liquid-Rigid Interaction Problems

K Sugiyama[1,2], H Okubo[1], R Imahoko[3], J Sakakibara[3] and S Takagi[4]

[1] Graduate School of Engineering Science, Osaka University, 1-3 Machikaneyama, Toyonaka, Osaka, 560-8531, Japan
[2] Advanced Center for Computing and Communication, RIKEN, 2-1 Hirosawa, Wako, Saitama, 351-0106, Japan
[3] Department of Mechanical Engineering, Meiji University, 1-1-1 Higashimita, Tama-ku, Kawasaki, Kanagawa, 214-8571, Japan
[4] School of Engineering, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan
kazuyasu.sugiyama@me.es.osaka-u.ac.jp

Abstract. A fixed-mesh approach has been developed to facilitate predicting a certain class of gas-liquid-rigid interaction problems. All the basic equations are discretized on a Cartesian grid in a finite-difference manner. The Volume-Of-Fluid and Boundary Data Immersion methods are employed to treat the gas-liquid and fluid-rigid interfaces, respectively. A hybrid OpenMP-MPI approach is adopted for parallel computing. The developed code is validated through comparisons with experiments of an oil-air flow driven by a rotating disk with holes. The simulated results demonstrate the capability in capturing the velocity distributions.

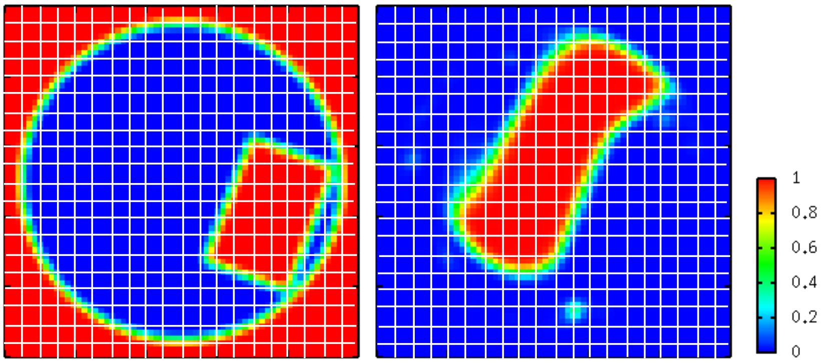
1. Introduction

Gas-liquid flows driven by rotating rigid objects appear in many engineering applications. The interplay among centrifugal, buoyancy, viscous and surface tension forces gives rise to a rich behaviour involving the gas-liquid and fluid-rigid interface motions. Large-scale simulations of multiphase flows are expected to be useful for complementing experiments and gaining insight into the dynamics. It is desirable that the numerical method is efficient and robust for massively parallel computing.

Direct numerical simulation of interfacial flows is often formidable mainly due to the spatiotemporal change in moving and deforming interface and the high degree of freedom. There are currently several approaches classified with respect to the numerical treatment how the kinematic and dynamic interactions are coupled on the interface. When addressing fluid engineering problems of practical applications with complex boundary, one has preferably employed a Lagrangian method using a finite element mesh. The numerical methods include Arbitrary Lagrangian Eulerian [1] and Deforming-Spatial-Domain/Stabilized Space-Time [2] approaches. If the body-fitted mesh is provided, the state-of-the-art Lagrangian approaches are satisfactory for achieving accurate predictions. However, for a system involving complex boundary, it requires considerable efforts to generate the high quality mesh and to reconstruct the mesh topology when the mesh element is added or deleted. Moreover, in parallel computing, one often encounters nontrivial issues how to keep the computational-load balance of each compute node in the domain decomposition, how to optimize the data communication between the Eulerian and Lagrangian frames, and so on. In the present study, we

Published under licence by IOP Publishing Ltd

focus on a fixed-mesh approach, which releases numerical simulations of gas-liquid-rigid motions from the mesh generation/reconstruction procedure and facilitates a prediction of a certain class of the problems. For the interface recognition, we use the rigid and gas volume fractions (see figure 1). Note that such an approach has been already applied to practical problems [3]. We implement novel methods to realize efficient and robust simulations of the gas-liquid flow interacting with the rigid body [4].



(a) rigid volume fraction (b) gas volume fraction

Figure 1. Schematic illustration explaining the interface recognition in the present fixed-mesh approach. The rigid and gas volume fractions distinguish among the rigid, gas and liquid phases. The contour at 0.5 indicates the interface.

In direct numerical simulations of multiphase flows, fixed-mesh approaches known as interface-capturing methods (for instance, Volume-Of-Fluid (VOF) [5], level set [6], and phase field [7, 8] methods) are often employed. One set of governing equations is written over the whole flow field, referred to as a one-fluid formulation [9]. For the gas-liquid flows, we focus on the VOF method, where the volume fraction of an each phase is updated as a solution to the advection equation. The VOF method exactly guarantees the numerical conservation of the volume fraction as long as the advection term is discretized in a flux-divergence form. However, since the advection equation includes no diffusion term, a special treatment is required to prevent the numerical instability and to maintain the sharp interface, simultaneously. The numerical methods for maintaining the sharp interface include Simple Line Interface Calculation (SLIC) [10], Piecewise Linear Interface Calculation (PLIC) [11], and Multi-dimensional Tangent of Hyperbola INterface Capturing (MTHINC) [12] methods. We employ the MTHINC method, which imposes a grid-scale smoothness on the volume fraction profile across the interface, and relieves the singularity problem due to the jump thereof. Further, in view of computational efficiency, the MTHINC procedure and the one-fluid formulation readily exploit vector processing and reduce load imbalance in parallel computing.

To address fluid-rigid interaction problems, the Immersed Boundary (IB) method [13, 14] has been frequently used as a fixed-mesh approach. In the IB method and a number of methods derived therefrom, the dynamic effect of the rigid body on the fluid flow is considered as an artificial body force in the fluid momentum equation to impose the no-slip boundary condition approximately on the interface. Although many options for the forcing have been explored so far, the body force expression itself makes it difficult to couple the pressure and the forcing and to satisfy the mass conservation, simultaneously. When a standard numerical algorithm for incompressible fluid flows is used in time-marching, the solenoidal condition of the velocity vector or the kinematic condition on the boundary is likely to be deteriorated. In the present study, we employ the Boundary Data Immersion (BDI) method [15], which solves a meta-equation written over the entire domain in a temporally discretized form instead of the fluid equation with the artificial forcing. The BDI method ensures exact enforcement of the solenoidal condition and the kinematic condition, and maintain physically consistent behaviour near the interface with a grid-scale smoothness.

To predict oil-air-rigid motions driven by a rotating disk, we have developed a fixed-mesh approach and made a parallelized code based on a hybrid OpenMP-MPI programming model. In this paper, we present the equation set and the numerical methods. Then, we explain an experimental setup, and make a comparison between the experiment and the direct numerical simulation, and discuss the validity of the numerical method and the applicability to large-scale simulations for practical systems.

2. Basic equations

Let us consider an three-dimensional incompressible domain D. It consists of liquid D_L, gas D_G and rigid D_R domains. We shall restrict out attention to systems such as the rigid domain D_R composed of a body D_{rot} rotating at a constant angular velocity Ω (here, the *z* axis is taken to be identical to the rotation axis) and a closed casing D_{rest} at rest. Hereafter, the suffices L, G and R denote liquid, gas, rigid phases, respectively, and the suffices "rot" and "rest" denote the rotating and resting portions, respectively. The mass conservation equation is

$$\nabla \cdot \mathbf{u} = 0, \ \mathbf{x} \in \mathrm{D}, \tag{1.1}$$

where **u** is the velocity vector, which is monolithically defined over the entire domain.

We introduce a volume fraction ϕ_k of phase *k* inside a computational cell. The following relation holds among the phases.

$$\phi_\mathrm{L} + \phi_\mathrm{G} + \phi_\mathrm{R} = 1, \ \ \phi_\mathrm{R} = \phi_\mathrm{rot} + \phi_\mathrm{rest}. \tag{1.2}$$

The volume fractions ϕ_G, ϕ_{rot} and ϕ_{rest} obey

$$\frac{\partial \phi_\mathrm{G}}{\partial t} + (\mathbf{u} \cdot \nabla)\phi_\mathrm{G} = 0, \tag{1.3}$$

$$\phi_\mathrm{rot}(\mathbf{x},t) = \phi_\mathrm{rot}(\mathbf{X}, t=0), \ \phi_\mathrm{rest}(\mathbf{x},t) = \phi_\mathrm{rest}(\mathbf{x}, t=0), \tag{1.4}$$

where *t* is the time. The initial location **X** of the rotating body is

$$\mathbf{X} = \mathbf{e}_x(x\cos\Omega t + y\sin\Omega t) + \mathbf{e}_y(-x\sin\Omega t + y\cos\Omega t) + \mathbf{e}_z z, \tag{1.5}$$

where **e** denotes the unit vector, and $\mathbf{x} = \mathbf{e}_x\, x + \mathbf{e}_y\, y + \mathbf{e}_z\, z$. On the basis of a one-fluid formulation [9] using a continuum surface force model [16], the momentum equation are written over the liquid and gas phases:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u}\right) = -\nabla p + \nabla \cdot \boldsymbol{\tau} - \gamma\kappa\nabla\phi_\mathrm{G} + \rho\mathbf{g}, \ \mathbf{x} \in \mathrm{D_L} \cup \mathrm{D_G}, \tag{1.6}$$

where ρ (= $(1-\phi_G)\,\rho_L + \phi_G\,\rho_G$) is the mixture density, *p* is the pressure, γ is the surface tension, and **g** is the acceleration of gravity. The viscous stress tensor **τ** and the curvature κ of the interface are respectively

$$\boldsymbol{\tau} = \mu\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^\mathrm{T}\right), \ \kappa = \nabla \cdot \mathbf{n}, \tag{1.7}$$

where μ (= $(1-\phi_G)\,\mu_L + \phi_G\,\mu_G$) is the mixture viscosity, and **n** is the unit normal vectors given by

$$\mathbf{n} = \frac{\mathbf{m}}{|\mathbf{m}|}, \ \mathbf{m} = \nabla \phi_{\mathrm{G}}. \tag{1.8}$$

The fluid momentum equation (equation (1.6)) is not applied to the rigid domain $\mathrm{D_R} (= \mathrm{D_{rot}} \cup \mathrm{D_{rest}})$. Instead, the velocity vector is prescribed as

$$\begin{cases} \mathbf{u} = \mathbf{U}_{\mathrm{rot}}, & \mathbf{x} \in \mathrm{D_{rot}}, \\ \mathbf{u} = 0, & \mathbf{x} \in \mathrm{D_{rest}}, \end{cases} \tag{1.9}$$

where $\mathbf{U}_{\mathrm{rot}} = -\mathbf{e}_x \Omega y + \mathbf{e}_y \Omega x$. Note that although the equations for the fluid and rigid motions are separately formulated as equations (1.6) and (1.9) for the respective domains, a discretized equation monolithically for both the motions is solved over the entire domain as explained in section 3.

3. Numerical methods

In the direct numerical simulation, the basic equations are solved by means of finite difference method using a regular Cartesian mesh. We follow a conventional staggered cell arrangement. The spatial derivatives are approximated by the second-order central differences, except for the advection term in equations (1.3) and (1.4) and **m** in equation (1.8).

As a solution to equation (1.3), the gas volume fraction ϕ_{G} (the VOF function) is advected. We employ the MTHINC method [12], in which a continuous sigmoid function (a hyperbolic tangent function) represents the interface in an algebraic form. The procedure enforces a grid-scale smoothness on the jump in the VOF profile across the interface, and avoids the numerical instability and diffusion. When updating the VOF function at a certain grid point, we refer only to the adjacent grid points, and need no iterative process. Thus, in view of computational efficiency, the MTHINC method makes it easily possible to reduce a computational-load imbalance and a data communication in parallel computing. Further, reconsidering the algorithm in [12], we replaced “if” statements with mask processings for the procedure differed by the direction normal to the interface, improving the vector processing efficiency.

To evaluate the unit normal vector **n**, we follow the Youngs approach [11], in which **m** in equation (1.8) is evaluated at the cell apex, and then it is interpolated onto the definition point of **n**. As in equation (1.4), $\phi_{\mathrm{rot}}(\mathbf{x}, t)$ is determined from the initial (t = 0) distribution of ϕ_{rot} in a semi-Lagrangian manner. A bilinear interpolation is applied to computing the value at the initial location **X** (in equation (1.5)) from the surrounding grid points.

The time-stepping algorithm to update the velocity and pressure fields at the (N+1)-th time level from the n-th time level is based on the SMAC method [17], which consists of unprojection and projection steps. From the known velocity vector $\mathbf{u}^N$ and VOF function ϕ_{G}^N (here, the superscript N stands for the N-th time level), the new VOF function ϕ_{G}^{N+1} is identified. As in equations (1.2) and (1.4), $\phi_{\mathrm{R}}^{N+1}(\mathbf{x})$ is given by $\phi_{\mathrm{rot}}^{N+1}(\mathbf{x})$ and $\phi_{\mathrm{rest}}^{0}(\mathbf{x})$, and $\phi_{\mathrm{rot}}^{N+1}(\mathbf{x})$ is determined by the projection to the initial value $\phi_{\mathrm{rot}}^{0}(\mathbf{X})$. We employ the BDI method [15], which allows us to monolithically treat the liquid-gas-rigid motions (equations (1.6) and (1.9)) over the entire domain. In the unprojection step, together with the updated volume fractions ϕ_{G}^{N+1}, $\phi_{\mathrm{rot}}^{N+1}$ and ϕ_{R}^{N+1}, the first-order explicit and implicit Euler schemes are applied to finding the unprojected velocity vector $\mathbf{u}^*$ as a solution to

$$\mathbf{u}^* = \phi_{\text{rot}}^{N+1}\mathbf{U}_{\text{rot}} + (1-\phi_{\text{R}}^{N+1})\left\{\mathbf{u}^N + \Delta t\left(-(\mathbf{u}^N\cdot\nabla)\mathbf{u}^N + \mathbf{g} + \frac{-\nabla p^N + \nabla\cdot\boldsymbol{\tau}^* - \gamma\kappa^{N+1}\nabla\phi_{\text{G}}^{N+1}}{\rho^{N+1/2}}\right)\right\}, \tag{2.1}$$

where Δt denotes the time increment, and

$$\rho^{N+1/2} = \frac{\rho^N + \rho^{N+1}}{2}, \boldsymbol{\tau}^* = \mu^{N+1}\left(\nabla\mathbf{u}^* + (\nabla\mathbf{u}^*)^{\text{T}}\right). \tag{2.2}$$

In the projection step, the pressure and the solenoidal velocity vector at the (N+1)-th time level are updated as

$$p^{N+1} = p^N + \psi,\ \ \mathbf{u}^{N+1} = \mathbf{u}^* - \Delta t\left(\frac{1-\phi_{\text{R}}^{N+1}}{\rho^{N+1/2}}\right)\nabla\psi, \tag{2.3}$$

where ψ is the incremental pressure as a solution to

$$\nabla\cdot\left\{\left(\frac{1-\phi_{\text{R}}^{N+1}}{\rho^{N+1/2}}\right)\nabla\psi\right\} = \frac{\nabla\cdot\mathbf{u}^*}{\Delta t}. \tag{2.4}$$

For parallel computing, an OpenMP-MPI hybrid programming model is adopted. The kernel code is implemented into an object-oriented framework V-Sphere [18], which includes class libraries to facilitate the software development of MPI parallel programs. A domain decomposition is applied to the entire computational domain D given as a set of regularly-divided rectangular domains, that minimizes the computational-load imbalance. Buffer arrays for the adjacent communication are allocated, and the data to/from the contiguous buffer are packed/unpacked. Equations (2.1) and (2.4) are iteratively solved by means of four-colour SOR and BiCG methods, respectively, which are suited to hybrid parallel computing.

To describe a multi-component geometry, the fixed mesh approach uses voxel data instead of boundary-fitted grids, and makes it easily possible to realize a massively parallel computing. As a preprocessing, utilizing an available software V-Xgen [19] converting from CAD data in a STL format into voxel data, we obtain the initial rigid volume fractions $\phi_{\text{rot}}^0(\mathbf{X})$ and $\phi_{\text{rest}}^0(\mathbf{X})$ of the rotating and resting portions.

4. Experimental setup

Experiments of an oil-air-rigid interaction problem are conducted using a container and a rotating disk. The system geometry is schematically illustrated in figure 2. The apparatus consists of resting and rotating parts made of acrylic resin for optical measurement. The resting part is a cylindrical container with the diameter of 270mm and the axial length of 360mm. The rotating part is a coaxial disk with the diameter of 250mm connected to a shaft. The disk has six circular holes. The container is placed in a way such as the cylinder axis pointing to the horizontal (z) direction. As shown in the right panel of figure 2, the position of z = 0 is located at the left fluid boundary in contact with the wall. The y axis is taken to be vertically downward.

Oil is poured into the container. The oil level in a stationary state is set to 12mm below the z axis. The oil-air flows are driven by the disk and shaft rotating at a angular speed Ω. The oil has the density of ρ_{L} = 0.89×10^3 kg/m^3 and the viscosity of μ_{L} = 67 mPa s. The oil-air surface tension is γ = 21 mN/m. Since the oil viscosity is dependent sensitively upon the temperature, it is important to keep the oil temperature constant. During the experiments, the oil temperature was monitored, and confirmed to be confined within a range between 22 and 25 (^{o}C).

The oil velocity field is measured by means of a Particle Image Velocimetry (PIV). The light source is a double pulse Nd-YAG laser (Nano-S PIV, 50mJ, 532nm, Litron Laser). A laser sheet is projected into the container via a cylindrical lens, and thereon time-serial images including tracer particles (FLUOSTAR, 15μm, Rhodamine B, EBM) are recorded using a high speed camera (CLB-B1922M-SC000, IMPERX). The synchronization between the camera shutters and the laser pulses is realized using a pulse generator (Model 9614, 4ch, Quantum Composers). The two-dimensional velocity distribution on the cross-section is determined by the PIV processing with time. For technical details on the procedure for gas-liquid two-phase flows, see [20, 21].

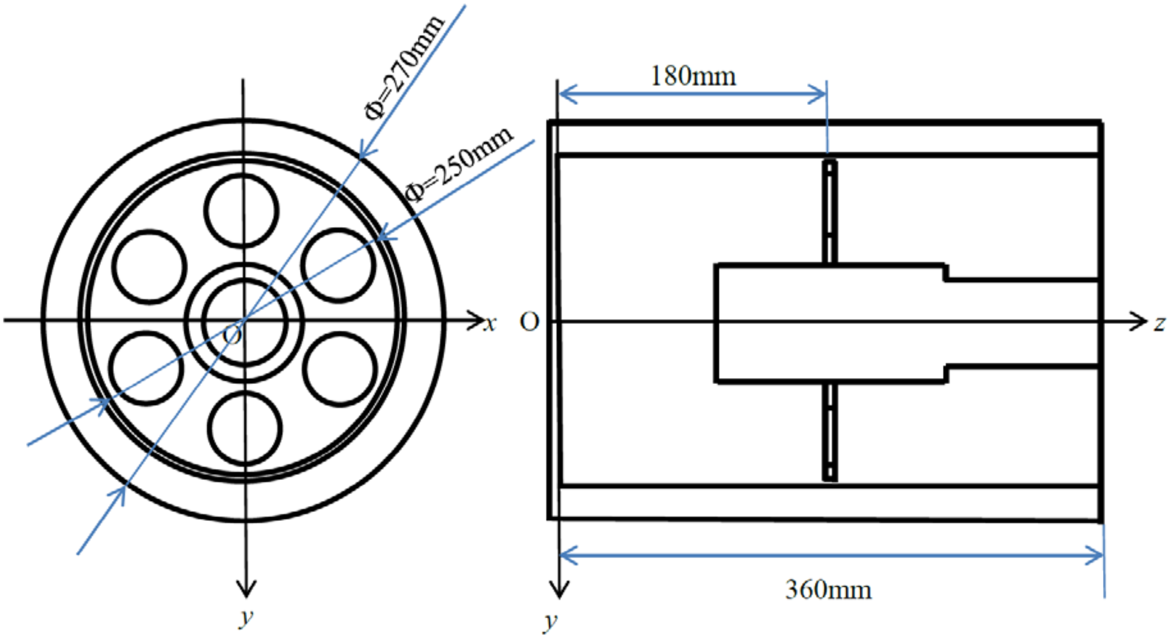


Figure 2. Schematic geometry of the container, the disk and the shaft

5. Results and discussion

The numerical simulations are performed with varying the angular speed Ω. The size of the computational domain is $L_x \times L_y \times L_z$ = 320mm×320mm×432mm, and the number of grid points is $N_x \times N_y \times N_z$ = 320×320×432. Hence, the grid size is Δx = 1mm. We use 128 compute nodes in the MPI parallel computation.

Figure 3 visualizes typical instantaneous distributions of oil at a fully developed state for various angular speeds Ω. The oil in contact with the disk moves along the disk rotating in the counterclockwise direction. The inclination of the gas-liquid interface from the horizontal plane is larger with increasing Ω because of the larger driving force. At the lower Ω (≤ 100rpm), the interface shape is nearly steady due to the continued forcing introduced by the rotating disk and shaft, and reveals weakly wavy motion due to the temporally-periodic forcing by the holes periodically bored on the disk. It is because the buoyancy effect is large enough to stabilize the system. At the higher Ω (≥ 200rpm), the oil raised by the circumferential edge of the disk reaches the top region of the container, and falls onto the horizontal oil surface. The interface shape reveals strongly unsteady motion. It is because the driving effect through the viscous stress near the disk and the centrifugal force effect due to the circumferential motion of the oil locally surpass the buoyancy effect, and thus the system is destabilized.

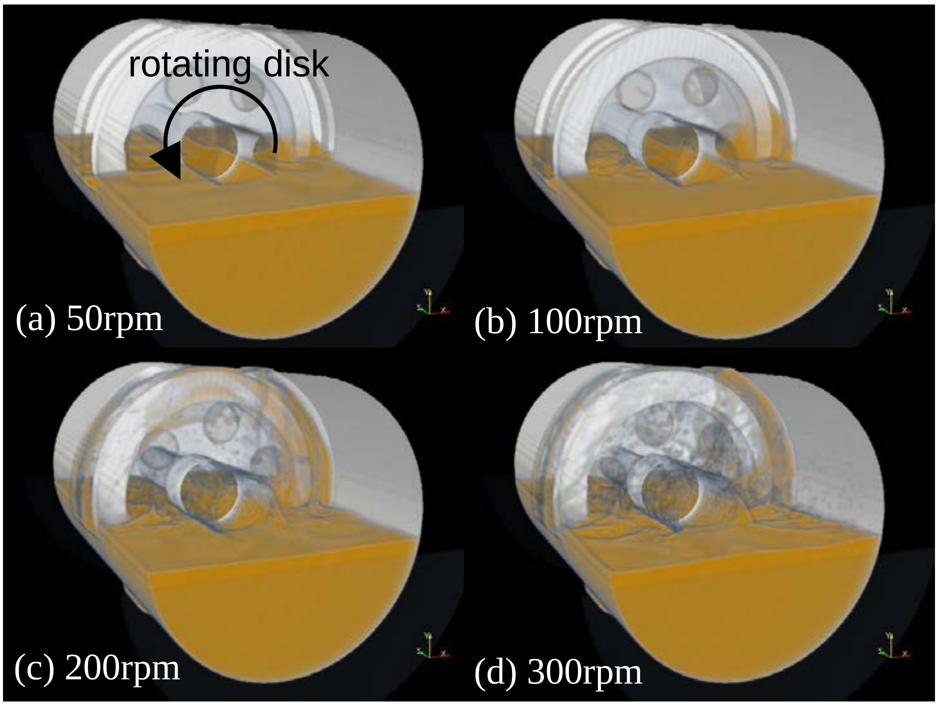
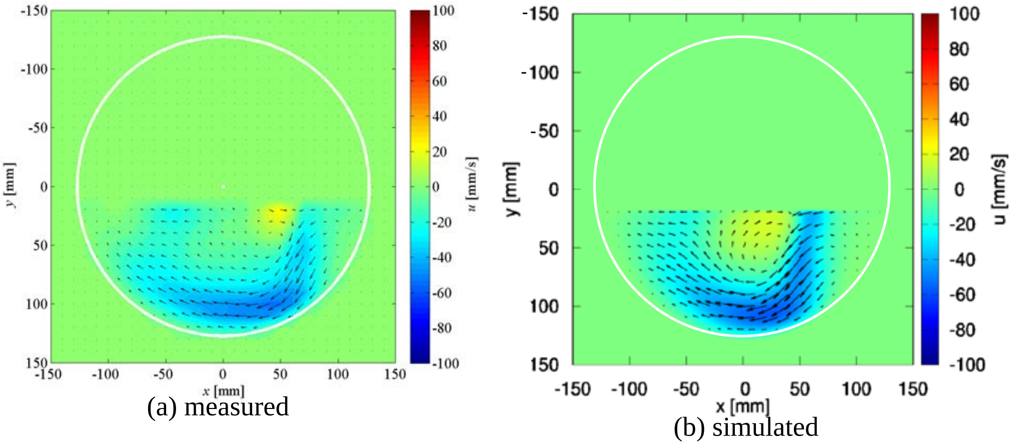


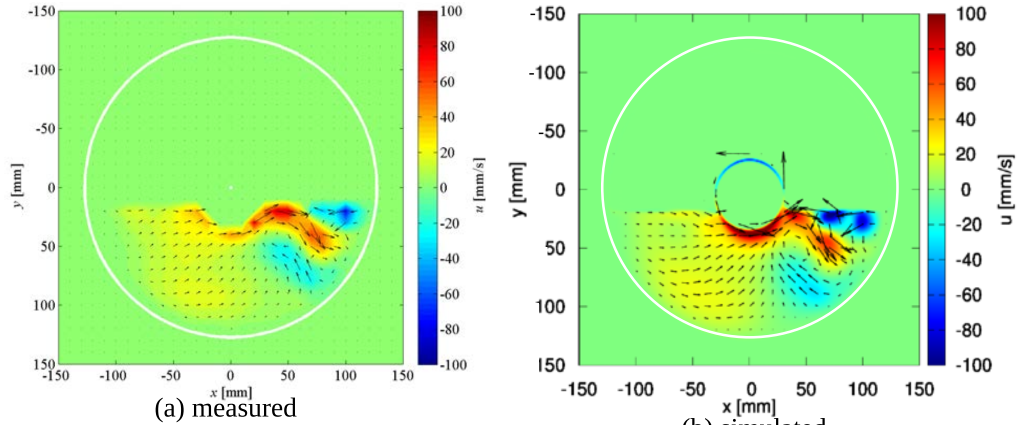
Figure 3. The simulated oil distributions for various angular speeds Ω

Figure 4 shows the velocity distributions of the oil on the cross-section z = 27mm at the angular speed of Ω = 50rpm. The left panel (figure 4(a)) shows the measured result, while the right panel (figure 4(b)) the simulated one. As shown in figure 4, the oil globally moves in the clockwise direction, which is opposite to that of the rotating disk. The three-dimensionality in the complicated flow structure is reflected on the difference in the flow direction. The local flow direction and magnitude predicted by the present fixed-mesh approach are in good agreement with those by the PIV. Figure 5 shows the velocity distribution at the angular speed of Ω = 50rpm (the same as figure 4), but the cross-section is on z ≈ 110mm. In figure 5(a) for the PIV measurement, the red region in the colour map indicates the jetting flow from the shaft, and the surrounding blue region the entrained flow. The present numerical approach (figure 5(b)) well reproduces such a complicated flow pattern quantitatively.



(a) measured (b) simulated

Figure 4. The time-averaged distribution of the velocity vector on the cross-section of z = 27mm at the angular speed of Ω = 50rpm. The colour map represents the magnitude of the x velocity component u.

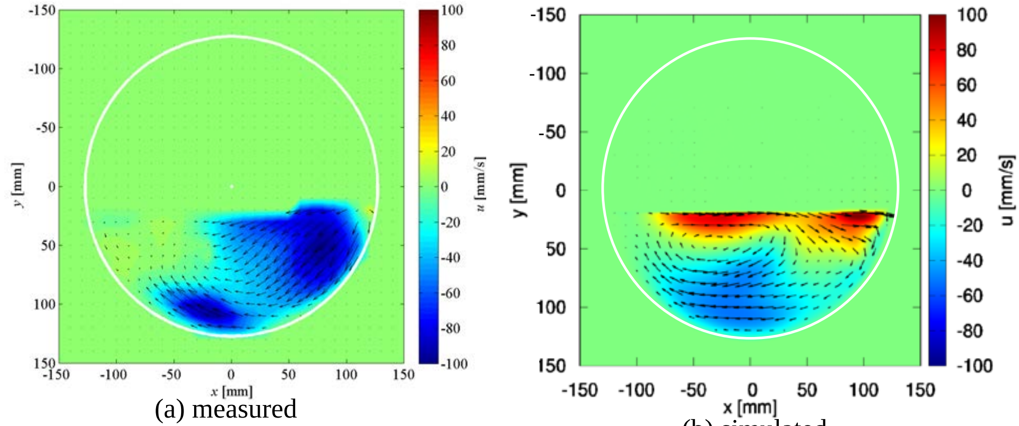


(a) measured

(b) simulated

Figure 5. The same as figure 4 but the cross-section of $z \approx 110$mm.

At the higher angular speed of Ω = 300rpm, at which the oil motion is strongly unsteady (see figure 3(d)), figure 6 shows the velocity distributions on the cross-section of z = 27mm. Unlike the nearly steady case (Ω = 50rpm, figure 4), the discrepancy between the measured and simulated distributions is clearly recognized. To resolve this problem, further joint research combining numerics and experiment is needed in consideration of the sampling time and the accuracy in the simulation and/or measurement. Note that the unsteady effect of the falling oil impacting on the oil surface (y = –12mm) may be preferentially obvious in the upper oil region, where the discrepancy is actually considerable in figure 6. By contrast, the lower oil region is not much affected by the unsteady effect, and therein the flow direction and magnitude are reasonably captured by the present simulation.



(a) measured

(b) simulated

Figure 6. The same as figure 4 but the angular speed of Ω = 300rpm.

6. Conclusion

A numerical method based on a fixed-mesh approach has been developed to facilitate a numerical solution to a gas-liquid-rigid interaction problem and to realize a massively parallel computation. In the paper, the applicability to larger-scale simulations for practical systems was demonstrated. To validate the developed code, experiments and numerical simulations of oil-air flows driven by a rotating disk container were performed for various angular speeds Ω. Even though the agreement became worse with increasing Ω, the present simulation could reasonably capture the velocity distribution experimentally obtained by the PIV.

Acknowledgments

Part of the results was obtained by using HOUKUSAI GreatWave at Advanced Center for Computing and Communication, RIKEN.

References

[1] Hirt C W Amsden A A and Cook J L 1974 An arbitrary Lagrangian-Eulerian computing method for all flow speeds *J Comput Phys* **14** 227
[2] Tezduyar T E Behr M and Liou J 1992 A new strategy for finite element computations involving moving boundaries and interfaces – The deforming-spatial-domain/space-time procedure: I The concept and the preliminary numerical tests *Comput Meth Appl Mech Engrg* **94** 339
[3] Arisawa H Nishimura M Imai H and Goi T 2014 Computational fluid dynamics simulations and experiments for reduction of oil churning loss and windage loss in aeroengine transmission gears *J Engrg Gas Turbine Power* **136** 092604
[4] Sugiyama K Niho T Takagi S and Fukui R 2015 A fixed-mesh approach for predicting gas-liquid flows driven by rotating objects *Proc of 13th Asian International Conference on Fluid Machinery* AICFM13-110
[5] Hirt C W and Nichols B D 1981 Volume of fluid (VOF) method for the dynamics of free boundaries *J Comput Phys* **38** 201
[6] Sussman M Smereka P and Osher S 1994 A level set approach for computating solutions to incompressible two-phase flow *J Comput Phys* **114** 146
[7] Wheeler A A Boettinger W J and McFadden G B 1992 A phase-field model for isothermal phase transitions in binary alloys *Phys Rev A* **45** 7424
[8] Jacqmin D 1999 Calculation of two-phase Navier-Stokes flows using phase-field modeling *J Comput Phys* **155** 96
[9] Tryggvason G Sussman M and Hussaini M Y 2007 Immersed boundary methods for fluid interfaces *Computational Methods for Multiphase Flow* eds A Prosperetti and G Tryggvason (Cambridge: Cambridge University Press) chapter 3
[10] Noh W F and Woodward P R 1976 SLIC (simple line interface calculation) *Lecture Notes in Phys* **59** 330
[11] Youngs D L 1982 Time-dependent multi-material flow with large fluid distortion *Numerical Methods for Fluid Dynamics* eds K W Morton and M J Baines (New York: Academic Press) pp 273–285
[12] Ii S Sugiyama K Takeuchi S Takagi S Matsumoto Y and Xiao F 2012 An interface capturing method with a continuous function: The THINC method with multi-dimensional reconstruction *J Comput Phys* **231** 2328
[13] Peskin C S 1972 Flow patterns around heart valves: a numerical method *J Comput Phys* **10** 252
[14] Peskin C S 2002 The immersed boundary method *Acta Numerica* **11** 479
[15] Weymouth G D and Yue D K P 2011 Boundary data immersion method for Cartesian-grid simulations of fluid-body interaction problems *J Comput Phys* **230** 6233
[16] Brackbill J U Kothe D B and Zemach C 1992 A continuum method for modeling surface tension *J Comput Phys* **100** 335
[17] Amsden A A and Harlow F H 1970 A simplified MAC technique for incompressible fluid flow calculations *J Comput Phys* **6** 322
[18] Ono K Tamaki T and Yoshikawa H 2009 Development of a framework for parallel simulators with various physics and its performance *Lecture Notes in Comput Sci Engrg* **67** 9
[19] Tawara T and Ono K 2007 Fast large scale voxelization using a pedigree *Proc of the 10th ISGG*

Conference on Numerical Grid Generation 2996

[20] Tokuhiro A Maekawa M Iizuka K Hishida K and Maeda M 1998 Turbulent flow past a bubble and an ellipsoid using shadow-image and PIV techniques *Int J Multiphase Flow* **24** 1383

[21] Fujiwara A Danmoto Y Hishida K and Maeda M Bubble deformation and flow structure measured by double shadow images and PIV/LIF *Exp in Fluids* **36** 157