

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/256918579>

A moving mesh finite volume interface tracking method for surface tension dominated interfacial fluid flow

ARTICLE *in* COMPUTERS & FLUIDS · FEBRUARY 2012

Impact Factor: 1.62 · DOI: 10.1016/j.compfluid.2011.11.003

CITATIONS

29

READS

179

2 AUTHORS:



[Zeljko Tukovic](#)

University of Zagreb

38 PUBLICATIONS 305 CITATIONS

[SEE PROFILE](#)



[Hrvoje Jasak](#)

University of Zagreb

62 PUBLICATIONS 1,781 CITATIONS

[SEE PROFILE](#)



A moving mesh finite volume interface tracking method for surface tension dominated interfacial fluid flow

Ž. Tuković^{a,*}, H. Jasak^{a,b}

^a University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture, Ivana Lučića 5, Zagreb, Croatia

^b Wikki Ltd., 33 Palmerston House, 60 Kensington Place, London W8 7PU, England, United Kingdom

ARTICLE INFO

Article history:

Received 25 March 2011

Received in revised form 14 October 2011

Accepted 8 November 2011

Available online 22 November 2011

Keywords:

Collocated finite volume method

Moving polyhedral mesh

Interface tracking

Surface tension

Multiphase fluid flow

PISO algorithm

Rhie–Chow interpolation

OpenFOAM

ABSTRACT

This paper describes a moving mesh interface tracking method implemented in OpenFOAM for simulating three-dimensional (3-D) incompressible and immiscible two-phase interfacial fluid flows with dominant surface tension forces. Collocated finite volume (FV) method is used for spatial discretisation of Navier–Stokes equations on moving polyhedral mesh. The mesh consists of two parts separated on interface. Fluid flow is solved on each mesh separately and coupling is accomplished in an iterative manner by enforcing the kinematic and dynamic condition at the interface. Surface tension force is calculated on arbitrary polygonal surface mesh with second order accuracy using a “force-conservative” approach. Arbitrary polyhedral mesh adapts to the time-varying shape of the interface using vertex-based automatic mesh motion solver which calculates the motion of internal points based on the prescribed motion of interface points by solving the variable diffusivity Laplace equation discretised using the finite element method. The overall solution procedure based on iterative PISO algorithm with modified Rhie–Chow interpolation is second-order accurate in space and time, as is confirmed by numerical experiments on small amplitude sloshing in a two-dimensional (2-D) tank, 3-D droplet oscillation and buoyant rise of a 3-D air bubble in water. Numerical results are found to be in excellent agreement with available theoretical and experimental results.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

Development of computational methods for numerical modeling of free-surface and interfacial fluid flow is a very active research area. The oldest and still the most popular methods use a fixed mesh to solve a single set of governing equations for the whole fluid flow, with different approaches to locating the interface between the phases. Among the many methods which belong to this category, one may mention the marker-and-cell method (MAC) [1,2], the volume of fluid method (VOF) [3–5], the level set method [6,7], the ghost fluid method [8] and a group of front tracking methods [9,10]. These methods are particularly attractive due to straight-forward handling of large deformation and changing topology of the interface. The main difficulty in using these methods is maintaining a sharp interface between the phases and computation of surface tension forces.

The moving mesh methods, also known as moving mesh interface tracking methods, use a separate boundary-fitted moving mesh for each phase. They offer potentially the highest accuracy since the interface between the phases is represented by the com-

putational boundary, allowing very accurate calculation of surface tension force and direct implementation of the kinematic and dynamic conditions at the interface without any smoothing of fluid phase properties. A major disadvantage of moving mesh methods is that they are limited to moderately deformed interfaces with constant topology. If one wishes to simulate large interface deformation or changing topology of the interface surface, some kind of global or local re-meshing would be necessary.

Although limited in their application, moving mesh interface tracking methods have an important role in the numerical analyses of fundamental multiphase flows such as motion of isolated bubbles and droplets, where strong dependence between interface shape and near surface flow exists, due to strong surface tension forces. Development of highly accurate moving mesh methods can also be justified by the fact that “numerically exact” solutions generated by such methods can be used for validation of fixed mesh methods.

One of the most famous studies performed by moving mesh methods which had a major impact on understanding of fundamental multiphase flows and subsequent development of computational methods for free-surface/interfacial fluid flows is that of Ryskin and Leal [11–13]. They studied steady buoyancy-driven motion of a single axisymmetric gas bubble by applying the finite-difference method on a structured orthogonal grid for numerical solution of

* Corresponding author.

E-mail addresses: Zeljko.Tukovic@fsb.hr (Ž. Tuković), h.jasak@wikki.co.uk (H. Jasak).

the Navier–Stokes equations expressed in terms of vorticity and stream function. Takagi et al. [14] used a finite-difference method on structured orthogonal mesh and the SIMPLER iterative solution procedure to simulate unsteady rise of a single axisymmetric bubble in quiescent liquid. Recently, Yang and Prosperetti [15] developed a moving mesh interface tracking method for axisymmetric unsteady free-surface flow calculation which has second-order accuracy in both space and time. In their methods structured orthogonal staggered mesh is used with finite-difference projection fractional step method to solve the primitive variable Navier–Stokes equations. Second order accuracy of the method is proved by performing numerical experiment on various forms of unsteady axisymmetric gas bubble motion.

All moving mesh interface tracking procedures mentioned above are restricted to axisymmetric or 2-D calculations. There exists only a limited number of studies where 3-D calculations were performed to analyse motion of bubbles or droplets using moving mesh methods. Perhaps the most impressive example is the work done by Schmidt and collaborators [16–18]. They developed a 3-D moving mesh interface tracking method where a staggered tetrahedral mesh is used with a FV discretisation and exact fractional step method to simulate motion, break up and coalescence of droplets and ligaments. This is a good example of moving mesh methods in combination with local mesh adaptation used to simulate multiphase flows with sharp interphase boundary, large interface deformation and changing surface topology.

This paper presents a 3-D moving mesh finite volume interface tracking method implemented in OpenFOAM. OpenFOAM [19] is an Open Source object-oriented C++ library for numerical simulations in continuum mechanics which provides second-order accurate FV discretisation with polyhedral mesh support, second-order discretisation in time, efficient linear system solvers and support for massively parallel computing. With the ultimate goal of developing a second-order accurate 3-D moving mesh interface tracking method with polyhedral mesh support, several extensions of the code described in this paper are made. A vertex-based mesh motion solver with polyhedral cell support is developed to calculate the motion of internal points based on the prescribed motion of interface (boundary) points. A collocated FV discretisation is extended to a moving mesh by taking into account the geometric conservation law in such a way to preserve accuracy in time established on a fixed mesh [20–22]. The interface tracking procedure uses the approach proposed by Muzafertija and Perić [23], where the interface is defined by boundary mesh faces and control points in order to preserve its smoothness during the calculation. Enforcement of the kinematic and dynamic conditions at the interface is performed in a semi-implicit manner within an iterative solution procedure. The dynamic condition at the interface requires calculation of derivatives of the velocity field along the interface. For this purpose, a counterpart of the finite volume method intended for discretisation on curved surface mesh is implemented. We refer to this discretisation procedure as the finite area method (FAM) [24] and part of it will be described in this paper.

The main concern during the implementation of the dynamic condition was how to implement surface tension force. Common practice in moving mesh methods is to apply the surface tension force through the product of mean curvature and the surface tension coefficient in pressure jump condition, where the mean curvature is usually calculated using a surface fitting procedure [25,17]. We have decided to use the novel “force-conservative” approach, where zero net surface tension force on any closed surface is satisfied exactly (see for example [26]). Special attention is paid to the accuracy of the local surface tension force calculation in order to reduce numerical noise responsible for parasitic currents around the interface [25].

The incompressible fluid flow model discretised on moving collocated mesh is solved using the iterative PISO algorithm [27] where the Rhie–Chow momentum interpolation method [28] is employed to evaluate cell face velocity in order to eliminate the checkerboard pressure field effect. Here we came across a limitation of the original Rhie and Chow interpolation connected with the application of very small time steps (due to semi-implicit treatment of surface tension forces). It is known that the original Rhie and Chow interpolation is time step size dependent and that it could fail to prevent the appearance of a checkerboard pressure field if a very small time step size is used [29–33]. Yu et al. [33] proposed a remedy for this problem on a fixed mesh and we have extended their solution, obtaining time step size independent Rhie and Chow interpolation on a moving mesh.

The remainder of the paper is organised as follows. In the next section, the governing equations for the two-fluid system with a sharp interface are described. The description of the FV discretisation for governing equations on a moving computational mesh is followed by description of the interface tracking solution procedure. The performance of the method is presented on three test cases and the paper is concluded with a summary.

2. Mathematical model

In this study, a two-phase fluid flow with a sharp interface is simulated using the FV method and a moving computational mesh. On each phase, one defines a separate computational mesh which moves and deforms according to interface motion. The mathematical model governing isothermal flow of incompressible fluid is solved separately on each moving mesh and coupling is accomplished by enforcement of proper boundary conditions at the interface.

Isothermal flow of incompressible Newtonian fluid inside an arbitrary volume V bounded by a closed moving surface S is governed by the mass and linear momentum conservation laws:

$$\oint_S \rho \mathbf{n} \cdot \mathbf{v} \, dS = 0, \tag{1}$$

$$\frac{d}{dt} \int_V \rho \mathbf{v} \, dV + \oint_S \mathbf{n} \cdot \rho (\mathbf{v} - \mathbf{v}_s) \mathbf{v} \, dS = \oint_S \mathbf{n} \cdot (\mu \nabla \mathbf{v}) \, dS - \int_V \nabla p \, dV, \tag{2}$$

where \mathbf{n} is the outward pointing unit normal on S , ρ is the fluid density, \mathbf{v} is the fluid velocity, \mathbf{v}_s is the velocity of surface S , μ is the dynamic viscosity of a fluid and p is the dynamic pressure obtained by subtracting hydrostatic pressure, $\rho \mathbf{g} \cdot \mathbf{r}$, from the absolute pressure, where \mathbf{g} is the gravitational acceleration and \mathbf{r} is the position vector. The above mathematical model, valid for arbitrary moving volume, is obtained from the corresponding material volume model using the Reynolds’ transport theorem. The relationship between the rate of change of the volume V and the velocity \mathbf{v}_s is defined by the *geometric (space) conservation law* (GCL, see [34,20]):

$$\frac{d}{dt} \int_V dV - \oint_S \mathbf{n} \cdot \mathbf{v}_s \, dS = 0. \tag{3}$$

If fluid phases are immiscible, fluid flow Eqs. (1) and (2) can be applied for each phase separately, while on the interface the proper boundary conditions must be used. Relation between fluid velocities on the two sides of the interface is determined by the *kinematic condition* [35], which states that the velocity must be continuous across the interface:

$$\mathbf{v}_A = \mathbf{v}_B, \tag{4}$$

where \mathbf{v}_A and \mathbf{v}_B are the fluid velocities at the two sides of the interface.

The dynamic condition follows from the momentum conservation law and states that forces acting on the fluid at the interface are in equilibrium. The tangential force balance yields a relation between normal derivative of tangential velocity on the two sides of the interface:

$$\mu_B(\mathbf{n} \cdot \nabla \mathbf{v}_t)_B - \mu_A(\mathbf{n} \cdot \nabla \mathbf{v}_t)_A = -\nabla_s \sigma - (\mu_B - \mu_A)(\nabla_s v_n), \quad (5)$$

where \mathbf{n} is the unit normal vector on the interface which points from fluid A to fluid B, $\mathbf{v}_t = (\mathbf{I} - \mathbf{nn}) \cdot \mathbf{v}$ is the tangential velocity component, $\nabla_s = \nabla - \mathbf{nn} \cdot \nabla$ is the surface gradient operator, σ is the surface tension coefficient and $v_n = \mathbf{n} \cdot \mathbf{v}$ is the normal velocity component at the interface. A non-zero gradient of the surface tension coefficient $\nabla_s \sigma$ can occur for example due to non-uniform distribution of surfactants at the interface or due to presence of a temperature gradient. In this study, the surface tension coefficient is assumed constant but the numerical procedure is described for the general case of a non-uniform surface tension coefficient.

From the normal force balance, the pressure jump across the interface is calculated:

$$p_B - p_A = \sigma \kappa - 2(\mu_B - \mu_A) \nabla_s \cdot \mathbf{v}, \quad (6)$$

where $\kappa = -\nabla_s \cdot \mathbf{n}$ is twice the mean curvature of the interface. The second term on the right hand side of Eq. (6) represents the jump of normal viscous force across the interface, expressed through surface divergence of interface velocity [36].

3. Numerical method

The mathematical model of fluid flow in its integral form is discretised in space using a second order accurate cell-centred unstructured FV method. Numerical integration of the model in time is performed using a second order accurate implicit method. The description of the discretisation procedure is divided into two parts: discretisation of the computational domain and equation discretisation.

3.1. Discretisation of the computational domain

The time interval is split into a finite number of time-steps Δt and the equations are solved in a time-marching manner. Computational space is divided into a finite number of convex polyhedral control volumes (CV) or cells bounded by convex polygons. Cells do not overlap and fill the spatial domain completely. Fig. 1 shows a polyhedral control volume V_P around the computational point P located in its centroid, face f , with area S_f , face unit normal vector \mathbf{n}_f

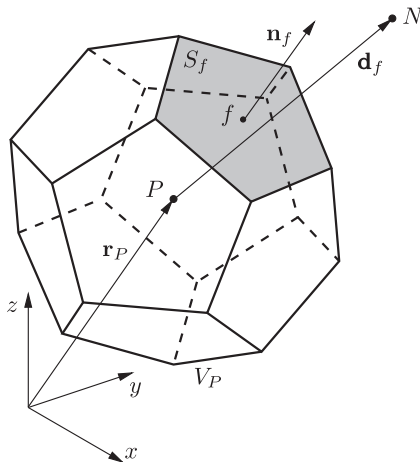


Fig. 1. Polyhedral control volume (cell).

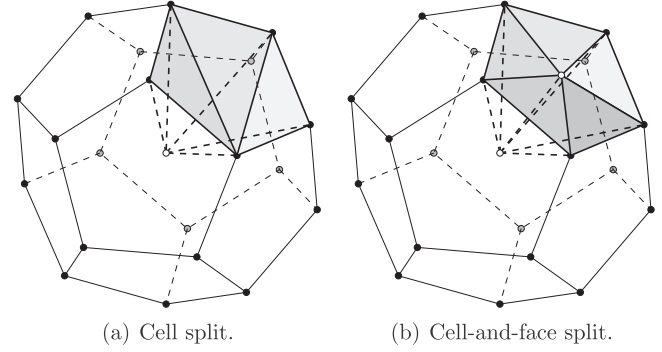


Fig. 2. Decomposing a polyhedral cell into tetrahedra.

and the centroid N of a neighbouring CV sharing the face f . Geometry of the CV is fully determined by the position of its vertices.

In surface tracking, the finite volume mesh needs to be adjusted to the time varying shape of the interface. The deforming mesh approach is used in this study where the internal CV vertices are moved based on the prescribed motion of the boundary vertices, while the topology of the mesh stays unchanged. The vertex-based automatic mesh motion solver [37,24] developed by the authors is used for mesh deformation. Here, displacement \mathbf{u} of the mesh points (vertices) is governed by the Laplace equation:

$$\nabla \cdot (\Gamma \nabla \mathbf{u}) = 0, \quad (7)$$

discretised on a tetrahedral finite element (FE) mesh using the Galerkin weighted residual FE method [38]. The diffusion coefficient Γ in Eq. (7) is inversely proportional to the square of distance from the moving boundary. This makes the mesh more rigid near moving boundaries and thus helps to preserve good mesh quality. The tetrahedral FE mesh is obtained by decomposition of polyhedral elements using one of the decomposition procedures shown in Fig. 2. In case of a complex polyhedral mesh the cell-and-face split is used despite introducing more computational points, as it produces better tetrahedron quality.

3.2. Discretisation of the mathematical model

According to the FV discretisation method, the surface integrals of an integral conservation equation are transformed into sums of face integrals which together with the volume integrals are approximated to second order accuracy by using the mid-point rule. Temporal discretisation is carried out by numerical integration of the governing equation in time from the old time instance t^o to the new time instance $t^n = t^o + \Delta t$ using an implicit three-level second order scheme [29] referred to as the backward scheme.¹ The fully discretised counterpart of the momentum Eq. (2) for the moving control volume V_P reads:

$$\begin{aligned} & \rho_P \frac{3\mathbf{v}_P^n - 4\mathbf{v}_P^o + \mathbf{v}_P^{oo}}{2\Delta t} + \sum_f (\dot{m}_f^n - \rho_f \dot{V}_f^n) \mathbf{v}_f^n \\ & = \sum_f \mu_f \mathbf{n}_f \cdot (\nabla \mathbf{v})_f^n S_f^n - (\nabla p)_P^n V_P^n, \end{aligned} \quad (8)$$

where the subscripts P and f represent the cell-centre and face-centre values and the superscripts n , o and oo represent values evaluated at the new time instance t^n and two previous time instance t^o and $t^{oo} = t^o - \Delta t$. The cell-face mass flux $\dot{m}_f^n = \rho_f \mathbf{n}_f \cdot \mathbf{v}_f^n S_f^n$ must satisfy the discretised mass conservation law, while the face volume flux \dot{V}_f^n must satisfy the discretised GCL. These unknown fluxes are

¹ The first order implicit Euler temporal discretisation scheme is used at the first time step since only one old time level is available.

treated as explicit in order to linearise the convection term in Eq. (8). The calculation of conservative fluxes will be described later in this section.

The face-centre values of most quantities are calculated using linear interpolation of the neighbouring cell-centre values:

$$\phi_f^n = \overline{(\phi_p^n)}_f = f_x^n \phi_p^n + (1 - f_x^n) \phi_N^n, \quad (9)$$

where ϕ is a general dependent variable and $f_x = \overline{fN/PN}$ is the interpolation factor, Fig. 1. Here, it is assumed that line PN intersects corresponding face in its centre. Otherwise, a “skewness” correction to the interpolated value could be applied [29]. In order to ensure boundedness while preserving second order accuracy, the face-centre velocity \mathbf{v}_f^n in the convection term of Eq. (8) is calculated by applying the Gamma interpolation scheme [39] which locally blends second order accurate linear interpolation with the unconditionally bounded upwind interpolation.

The face normal derivative of velocity, $\mathbf{n}_f^n \cdot (\nabla \mathbf{v})_f^n$, in the diffusion term of Eq. (8) is discretised as follows (see [40]):

$$\mathbf{n}_f^n \cdot (\nabla \mathbf{v})_f^n = \underbrace{\left| \Delta_f^n \right| \frac{\mathbf{v}_N^n - \mathbf{v}_P^n}{\mathbf{d}_{fu}^n}}_{\text{Orthogonal contribution}} + \underbrace{(\mathbf{n}_f^n - \Delta_f^n) \cdot (\nabla \mathbf{v})_f^n}_{\text{Non-orthogonal correction}}, \quad (10)$$

where $\Delta_f^n = \mathbf{d}_f^n / (\mathbf{d}_f^n \cdot \mathbf{n}_f^n)$, Fig. 1. The orthogonal contribution in Eq. (10) is treated as implicit, while the non-orthogonal correction is explicit.

The cell-centre gradient of pressure in Eq. (8) is calculated using the Gauss integral theorem in order to preserve physical representation of the pressure force at cell faces. All other cell-centre gradients are calculated using the least-squares fit [41,42]. This method produces a second-order accurate gradient irrespective of local mesh quality.

When Eq. (10) is substituted into Eq. (8) and Gamma convection discretisation scheme is applied, discretised momentum equation can be written in the form of a linear algebraic equation, which for cell P reads²:

$$a_p \mathbf{v}_p^n + \sum_N a_N \mathbf{v}_N^n = \mathbf{r}_p - (\nabla p)_p^n, \quad (11)$$

where the diagonal coefficient a_p , the neighbour coefficient a_N and the source term \mathbf{r}_p^n depend on the unknown velocity field due to explicit treatment of some terms in the discretised momentum equation (*i.e.* cell-face mass flux, face volume flux, non-orthogonal correction, *etc.*). For the analysis of the original Rhie–Chow interpolation and its deficiency related to the application of small time steps, only the contribution from the unsteady term in the diagonal coefficient and source term is important:

$$a_p = a_p^* + \frac{3\rho_p}{2\Delta t}, \quad (12)$$

$$\mathbf{r}_p = \mathbf{r}_p^* + \frac{2\rho_p V_p^{oo}}{V_p^n \Delta t} \mathbf{v}_p^{oo} - \frac{\rho_p V_p^{oo}}{2V_p^n \Delta t} \mathbf{v}_p^{oo}, \quad (13)$$

where superscript $*$ represents contribution from the convective and diffusive terms.

The mathematical model of the fluid flow is solved using a segregated solution procedure, where the momentum equation is decoupled from the pressure equation. The discretised pressure equation is obtained by combining the discretised momentum and continuity equations using the momentum interpolation method [29] as follows. Discretised continuity equation for cell P reads:

$$\sum_f \rho_f^n \mathbf{n}_f^n \cdot \mathbf{v}_f^n S_f^n = \sum_f \dot{m}_f^n = 0, \quad (14)$$

where \dot{m}_f^n is the cell-face mass flux. From the discretised momentum Eq. (11), one can express cell-centre velocity as follows:

$$\mathbf{v}_p^n = \frac{\mathbf{H}_p(\mathbf{v}^n)}{a_p} - \frac{1}{a_p} (\nabla p)_p^n, \quad (15)$$

where

$$\mathbf{H}_p(\mathbf{v}^n) = - \sum_f a_N \mathbf{v}_N^n + \mathbf{r}_p^n = \mathbf{H}_p^*(\mathbf{v}^n) + \frac{2\rho_p V_p^{oo}}{V_p^n \Delta t} \mathbf{v}_p^{oo} - \frac{\rho_p V_p^{oo}}{2V_p^n \Delta t} \mathbf{v}_p^{oo}. \quad (16)$$

According to the momentum interpolation method, the cell-face velocity needed in Eq. (14) can be expressed by mimicking Eq. (15) as follows:

$$\mathbf{v}_f^n = \left(\frac{\mathbf{H}}{a} \right)_f - \left(\frac{1}{a} \right)_f (\nabla p)_f^n, \quad (17)$$

where the terms $(\mathbf{H}/a)_f$ and $(1/a)_f$ are obtained by interpolating their counterparts in Eq. (15) written for the two cells sharing the face f . Usually this is done using Rhie–Chow interpolation [28], but other interpolation schemes can be found in the literature as well [43]. When Eq. (17) is substituted into Eq. (14), one obtains the discretised pressure equation for the cell P :

$$\sum_f \left(\frac{1}{a} \right)_f \mathbf{n}_f^n \cdot (\nabla p)_f^n S_f^n = \sum_f \mathbf{n}_f^n \cdot \left(\frac{\mathbf{H}}{a} \right)_f S_f^n, \quad (18)$$

where the normal pressure derivative term $\mathbf{n}_f^n \cdot (\nabla p)_f^n$ is discretised using Eq. (10). After the solution of the pressure Eq. (18), one can calculate divergence-free cell-face mass flux as follows:

$$\dot{m}_f^n = \rho_f \left[\mathbf{n}_f^n \cdot \left(\frac{\mathbf{H}}{a} \right)_f - \left(\frac{1}{a} \right)_f \mathbf{n}_f^n \cdot (\nabla p)_f^n \right] S_f^n. \quad (19)$$

According to the original Rhie–Chow interpolation, terms $(\mathbf{H}/a)_f$ and $(1/a)_f$ in Eqs. (18) and (19) are calculated by using the following expressions:

$$\left(\frac{\mathbf{H}}{a} \right)_f^{RC} = \overline{\left[\frac{\mathbf{H}_p(\mathbf{v}^n)}{a_p} \right]}_f, \quad \left(\frac{1}{a} \right)_f^{RC} = \overline{\left(\frac{1}{a_p} \right)}_f, \quad (20)$$

where $\overline{(\cdot)}_f$ denotes linear interpolation of the corresponding cell-centre values. It is well known that the original Rhie–Chow interpolation could fail to prevent a checkerboard pressure field if a very small time step is used [29,31–33]. Explanation for such behaviour can be provided by analysing limiting values of expressions (20) when time step size tends to zero:

$$\lim_{\Delta t \rightarrow 0} \left(\frac{\mathbf{H}}{a} \right)_f^{RC} = \frac{4}{3} \overline{(\mathbf{v}_p^{oo})}_f - \frac{1}{3} \overline{(\mathbf{v}_p^{oo})}_f, \quad \lim_{\Delta t \rightarrow 0} \left(\frac{1}{a} \right)_f^{RC} = 0, \quad (21)$$

where $\overline{(\mathbf{v}_p^{oo})}_f$ and $\overline{(\mathbf{v}_p^{oo})}_f$ are cell-face velocities from the two previous time steps obtained by interpolating corresponding cell-centre velocities. Since the two interpolated velocities do not satisfy the discretised continuity equation, solution of pressure equation will produce an unphysical pressure field in the considered limiting case. Another undesirable feature of the original Rhie–Chow interpolation is that the converged steady state cell-face mass flux calculated using Eq. (19) depends on the time step size [30,32,33]. This issue has manifested itself in our study through the consistency violation of the numerical method for small time steps. We have eliminated both of these two deficiencies by using the modified interpolation proposed by Yu et al. [33] for a fixed mesh. Hence, in our modified Rhie–Chow (mRC) interpolation, terms $(\mathbf{H}/a)_f$ and $(1/a)_f$ in Eqs. (18) and (19) are calculated using the following expressions:

² One should note here that Eq. (8) is divided by V_p^n .

$$\begin{aligned} \left(\frac{\mathbf{H}}{a}\right)_f^{mRC} &= \frac{\overline{\mathbf{H}_p^*(\mathbf{v}^n)}_f}{(\overline{a_p})_f} + \frac{\overline{\left(\frac{2\rho_p V_p^o}{V_p^o \Delta t}\right)}_f}{(\overline{a_p})_f} (\mathbf{v}_p^o)_f - \frac{\overline{\left(\frac{\rho_p V_p^{oo}}{2V_p^o \Delta t}\right)}_f}{(\overline{a_p})_f} (\mathbf{v}_p^{oo}), \\ \left(\frac{1}{a}\right)_f^{mRC} &= \frac{1}{(\overline{a_p})_f}, \end{aligned} \quad (22)$$

where $(\mathbf{v}_p^o)_f$ and $(\mathbf{v}_p^{oo})_f$ are cell-face velocities from the two previous time steps which satisfy discretised continuity equation on the corresponding previous mesh configurations:

$$(\mathbf{v}_p^o)_f = \overline{(\mathbf{v}_p^o)}_f + \mathbf{n}_f \left[\frac{\dot{m}_f^o}{\rho_f S_f} - \mathbf{n}_f \bullet \overline{(\mathbf{v}_p^o)}_f \right], \quad (23)$$

$$(\mathbf{v}_p^{oo})_f = \overline{(\mathbf{v}_p^{oo})}_f + \mathbf{n}_f \left[\frac{\dot{m}_f^{oo}}{\rho_f S_f} - \mathbf{n}_f \bullet \overline{(\mathbf{v}_p^{oo})}_f \right]. \quad (24)$$

One can show that pressure Eq. (18) with modified terms $(\mathbf{H}/a)_f$ and $(1/a)_f$ defined by expressions (22) is well defined even in the limiting case when time step size tends to zero and the cell-face mass flux obtained by Eq. (19) will not depend on time step size in steady state.

Demirdžić and Perić [20] showed that failure to satisfy the discretised geometric conservation law (DGCL) on moving meshes introduces errors in the form of artificial mass sources. Farhat and Geuzaine [22,21] showed that satisfaction of the DGCL is a necessary condition for any temporal discretisation scheme to preserve on moving grids the non-linear stability properties of its fixed-grid counterpart but they also showed that satisfaction of the DGCL is not a sufficient condition for a temporal discretisation scheme to preserve its order of time-accuracy on moving grids, as established on fixed grids.

In this study, cell-face volume fluxes \dot{V}_f are calculated to satisfy DGCL:

$$\frac{3V_p^n - 4V_p^o + V_p^{oo}}{2\Delta t} - \sum_f \dot{V}_f^n = 0, \quad (25)$$

where the three-level backward scheme is used for temporal discretisation. The difference between the cell volumes at consecutive time-levels can be decomposed as follows:

$$V_p^n - V_p^o = \sum_f \delta V_f^n, \quad (26)$$

where δV_f^n is the volume swept by the cell face f while moving from its old to its new position. Substitution of Eq. (26) into Eq. (25) yields:

$$\frac{1}{2\Delta t} \sum_f (3\delta V_f^n - \delta V_f^o) = \sum_f \dot{V}_f^n. \quad (27)$$

Hence, if the cell-face volume fluxes are calculated using the following expression:

$$\dot{V}_f^n = \frac{3}{2} \frac{\delta V_f^n}{\Delta t} - \frac{1}{2} \frac{\delta V_f^o}{\Delta t}, \quad (28)$$

the DGCL (25) will be satisfied identically and according to Geuzaine et al. [21] the backward scheme will preserve second order temporal accuracy on a moving mesh. The expression for calculation of cell-face volume fluxes which satisfy the GCL discretised using the first order implicit Euler temporal discretisation scheme are derived in [20].

The discretisation procedure described above deals with internal CV faces. Approximation of surface integrals at boundary faces depends on the prescribed boundary conditions. Implementation of the boundary conditions depending on their types is described in [40].

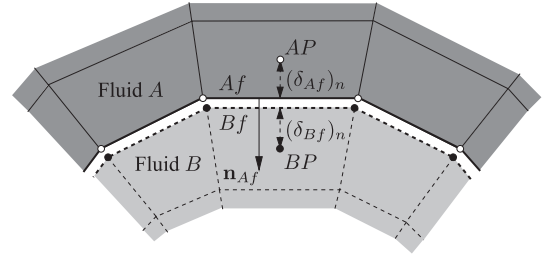


Fig. 3. Representation of the interface with the mesh boundary faces.

3.3. Interface tracking procedure

Numerical modelling of a two-phase fluid flow with a sharp interface is performed using a moving mesh interface tracking procedure. The computational mesh consists of two separate parts, where each part covers only one of the considered fluid phases. The two mesh parts are in contact over the two geometrically equal surfaces, S_A and S_B , at the boundary between the phases (the interface). Each surface is defined by a set of boundary faces, see Fig. 3, where each face Af on the surface S_A has a corresponding geometrically equal face Bf on the surface S_B . Matching of the two mesh parts at the interface is assumed in order to facilitate the presentation of the interface tracking method, and is not required in general. For non-matching meshes, a second-order inverse-distance weighted interpolation is used.

Coupling of flow equations between the phases is performed by applying adequate boundary conditions at the interface. At the side A of the interface which belongs to the denser phase³ ($\rho_A > \rho_B$), the dynamic pressure p_A and normal velocity derivative $\mathbf{n}_A \bullet (\nabla \mathbf{v})_A$ are specified, while at the side B the velocity \mathbf{v}_B and normal derivative of dynamic pressure $\mathbf{n}_B \bullet (\nabla p)_B$ are specified. These boundary conditions are updated at the beginning of each outer iteration using the following procedure⁴:

1. Tangential component of the interface velocity at the face Af is calculated using discretised counterpart of Eq. (5) with the kinematic condition (4) taken into account:

$$\begin{aligned} (\mathbf{v}_{Af})_t &= \frac{\frac{\mu_A}{(\delta_{Af})_n} (\mathbf{v}_{AP})_t + \frac{\mu_B}{(\delta_{Bf})_n} (\mathbf{v}_{BP})_t}{\frac{\mu_A}{(\delta_{Af})_n} + \frac{\mu_B}{(\delta_{Bf})_n}} \\ &+ \frac{(\nabla_s \sigma)_{Af} + (\mu_B - \mu_A) (\nabla_s v_n)_{Af}}{\frac{\mu_A}{(\delta_{Af})_n} + \frac{\mu_B}{(\delta_{Bf})_n}}, \end{aligned} \quad (29)$$

where $(\mathbf{v}_{AP})_t$ and $(\mathbf{v}_{BP})_t$ are the tangential components of velocity in the centroid of cells AP and BP (see Fig. 3), while $(\delta_{Af})_n$ and $(\delta_{Bf})_n$ are the normal distances from centroids of cells AP and BP to their corresponding faces Af and Bf . As one can note from Eq. (29), normal derivatives of tangential velocity in Eq. (5) are discretised using one sided first order approximation. The first term on the right hand side of Eq. (29) represents the “geometric harmonic mean” which is used in [17] for treating shear stress.

2. The normal velocity derivative specified at the face Af is calculated using updated interface velocity, as follows:

$$\mathbf{n}_{Af} \bullet (\nabla \mathbf{v})_{Af} = \mu_A \frac{(\mathbf{v}_{Af})_t - (\mathbf{v}_{AP})_t}{(\delta_{Af})_n} + \mu_A (\nabla_s \bullet \mathbf{v})_{Af} \mathbf{n}_f, \quad (30)$$

³ The denser phase is used as the phase A in order to achieve good convergence properties of the iterative coupling procedure without need for stabilisation.

⁴ In all expressions that follows superscript n is omitted since all quantities are evaluated at the current (new) time instance.

where $\mathbf{n}_{Af} = -\mathbf{n}_{Bf}$ is the unit normal of the face Af . Calculation of surface derivative terms in Eqs. (29) and (30) will be described later in this section.

3. According to the kinematic condition (4), the tangential velocity component specified on the face Bf is transferred from the face Af :

$$(\mathbf{v}_{Bf})_t = (\mathbf{v}_{Af})_t. \quad (31)$$

The normal velocity component specified on Bf is calculated from the condition of zero net mass flux, $(\dot{m}_{Bf} - \rho_B \dot{V}_{Bf}) = 0$, i.e.

$$(\mathbf{v}_{Bf})_n = \frac{\dot{V}_{Bf}}{S_{Bf}} \mathbf{n}_{Bf}, \quad (32)$$

where \dot{V}_{Bf} is the volumetric flux of the face Bf . Since the displacement of mesh points at the side B of the interface is equal to the displacement at the side A of the interface, $\mathbf{u}_{Bi} = -\mathbf{u}_{Ai}$, the same is valid for the volume fluxes of faces Af and Bf :

$$\dot{V}_{Bf} = \dot{V}_{Af}. \quad (33)$$

4. Value of the normal derivative of dynamic pressure specified at the face Bf is calculated from the surface-normal component of the momentum equation:

$$\mathbf{n}_{Bf} \bullet (\nabla p)_{Bf} = -\mathbf{n}_{Bf} \bullet \left(\frac{D\mathbf{v}}{Dt} \right)_{Bf}. \quad (34)$$

5. Value of the dynamic pressure specified at the face Af (see Fig. 3) is calculated using the condition (6) as follows:

$$p_{Af} = p_{Bf} - (\rho_A - \rho_B) \mathbf{g} \bullet \mathbf{r}_{Af} - (\sigma\kappa)_{Af} - 2(\mu_A - \mu_B)(\nabla_s \bullet \mathbf{v})_{Af}, \quad (35)$$

where \mathbf{r}_{Af} is the position vector of the face centre Af and p_{Bf} is the dynamic pressure at the face Bf obtained by extrapolation from the fluid B , using the specified normal derivative boundary condition for dynamic pressure. Calculation of the surface tension term $(\sigma\kappa)_{Af}$ will be described later in this section.

In general, at the end of an outer iteration, the net mass flux through the faces at the side A of the interface is different from zero, i.e.

$$(\dot{m}_{Af} - \rho_A \dot{V}_{Af}) \neq 0, \quad (36)$$

where \dot{m}_{Af} is the mass flux through the face Af and \dot{V}_{Af} is the volume face flux. In order to correct the net mass flux, the interface points must be moved to accomplish the following volume flux corrections:

$$\dot{V}'_{Af} = \frac{\dot{m}_{Af}}{\rho_A} - \dot{V}_{Af}. \quad (37)$$

The interface point displacement is calculated based on the procedure proposed in [23], where a control point Ac is attached above the centroid of each face Af at the side A of the interface as is shown in Fig. 4. The corrected position of interface points is calculated using the following procedure:

1. Calculate the volume $\delta V'_{Af}$ swept by the face Af on the way from the current to the corrected position in order to cancel the net mass flux:

$$\delta V'_{Af} = \frac{2}{3} \dot{V}'_{Af} \Delta t, \quad (38)$$

where \dot{V}'_{Af} is the volume flux correction for face Af calculated using Eq. (37). Eq. (38) is derived from Eq. (28).

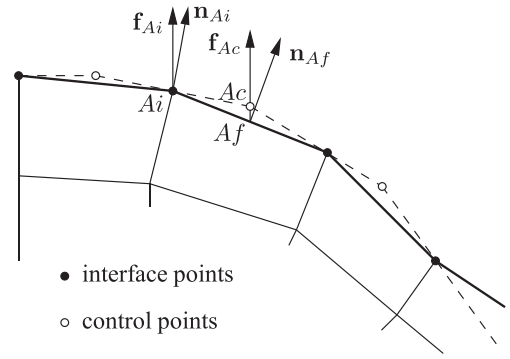


Fig. 4. Definition of the interface using control points.

2. Using the above, displacement of control points in the direction \mathbf{f}_{Ac} is

$$h_{Ac}^{prime} = \frac{\delta V'_{Af}}{S_{Af} \mathbf{n}_{Af} \bullet \mathbf{f}_{Ac}}, \quad (39)$$

where S_{Af} and \mathbf{n}_{Af} are the area and unit normal of the face Af in the current iteration and \mathbf{f}_{Ac} is the control point displacement direction. The new corrected position of control points is calculated according to the following expression:

$$\mathbf{r}_{Ac} = \mathbf{r}_{Ac}^p + h'_{Ac} \mathbf{f}_{Ac}, \quad (40)$$

where \mathbf{r}_{Ac}^p is the position vector of the control point Ac before the correction.

3. The interface mesh point Ai is moved up to a plane laid over the corresponding control points using the least squares method. The new position of interface mesh points is calculated using the following expression:

$$\mathbf{r}_{Ai} = \mathbf{r}_{Ai}^p + \frac{\mathbf{N}_{Ai} \bullet (\mathbf{p} - \mathbf{r}_{Ai}^p)}{\mathbf{N}_{Ai} \bullet \mathbf{f}_{Ai}} \mathbf{f}_{Ai}, \quad (41)$$

where \mathbf{f}_{Ai} is the displacement direction of the interface mesh point, \mathbf{N}_{Ai} is the unit normal vector on the plane:

$$\mathbf{N}_{Ai} = \frac{\mathbf{G}^{-1} \bullet \sum_{Ac} w_{Ac}^2 \mathbf{r}_{Ac}}{|\mathbf{G}^{-1} \bullet \sum_{Ac} w_{Ac}^2 \mathbf{r}_{Ac}|}, \quad (42)$$

and \mathbf{p} is the position of the point at the plane,

$$\mathbf{p} = \frac{\sum_{Ac} w_{Ac}^2 \mathbf{r}_{Ac}}{\sum_{Ac} w_{Ac}^2}. \quad (43)$$

Tensor \mathbf{G} is defined by the following expression:

$$\mathbf{G} = \sum_{Ac} w_{Ac}^2 \mathbf{r}_{Ac} \mathbf{r}_{Ac}. \quad (44)$$

In Eqs. (42)–(44), summation is performed over all control points common to the interface point Ai and weighting factor w_{Ac} is defined as the inverse distance from the control point Ac to the vertex Ai .

Unit vectors \mathbf{f}_{Ac} and \mathbf{f}_{Ai} which determine the displacement direction for the control points and the interface mesh points must be specified in advance for each time instance. Two variants are used in this study, depending on the considered interface tracking problem. The displacement direction can either be parallel with the gravity vector or with the surface normal vector from the previous time instance.

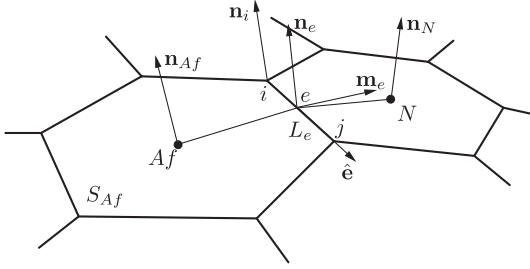


Fig. 5. Control area S_{Af} at the interface.

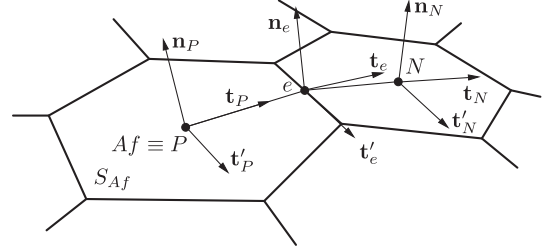


Fig. 6. Edge-based local orthogonal coordinate system whose axis are aligned with orthogonal unit vectors \mathbf{n} , \mathbf{t} and \mathbf{t}' , where vector \mathbf{t} is tangential to the geodesic line \overline{PeN} .

3.4. Calculation of surface derivatives

Surface divergence and surface gradient terms in Eqs. (35) and (30) are calculated using the surface Gauss' integral theorem, which, for a general tensorial quantity Φ defined on the surface S bounded by the closed line ∂S , reads as follows [44,45]:

$$\int_S \nabla_s \otimes \Phi \, dS = \int_{\partial S} \mathbf{m} \otimes \Phi \, dL - \int_S \kappa \mathbf{n} \otimes \Phi \, dS, \quad (45)$$

where \mathbf{n} is the unit normal vector on the surface S , \mathbf{m} is the unit bi-normal vector perpendicular to the line ∂S and tangential to the surface S and the operator \otimes represents a scalar, vector or tensor product operator.

In Eqs. (35) and (30) we seek the surface divergence of velocity $(\nabla_s \cdot \mathbf{v})_{Af}$ and the surface gradient of velocity $(\nabla_s \mathbf{v})_{Af}$ at the centroid of face Af . These quantities are calculated by discretising Eq. (45) on the control area S_{Af} (see Fig. 5) using basic principles of the finite volume (area) discretisation procedure. Surface divergence and surface gradient of velocity at the centroid of control area S_{Af} are calculated using the following expressions:

$$(\nabla_s \cdot \mathbf{v})_{Af} = \frac{1}{S_{Af}} \sum_e \mathbf{m}_e \cdot \mathbf{v}_e L_e - \kappa_{Af} \mathbf{n}_{Af} \cdot \mathbf{v}_{Af}, \quad (46)$$

$$(\nabla_s \mathbf{v})_{Af} = \frac{1}{S_{Af}} \sum_e \mathbf{m}_e \mathbf{v}_e L_e - \kappa_{Af} \mathbf{n}_{Af} \mathbf{v}_{Af}, \quad (47)$$

where surface integrals over the control area S_{Af} and line integrals over the control area edge e of length L_e are approximated using the mid-point rule. Subscript e implies the value of the variable in the middle of the edge e and summation is performed over all edges closing the face Af . The edge-centre unit bi-normal vector is calculated as follows:

$$\mathbf{m}_e = \hat{\mathbf{e}} \times \frac{\mathbf{n}_i + \mathbf{n}_j}{|\mathbf{n}_i + \mathbf{n}_j|}, \quad (48)$$

where $\hat{\mathbf{e}}$ is the unit vector parallel with the edge e , while \mathbf{n}_i and \mathbf{n}_j are the interface unit normal vectors at end points i and j of the edge e (see Fig. 5). Calculation of point normals \mathbf{n}_i and face curvatures κ_{Af} is described in the next subsection.

The edge-centre velocity \mathbf{v}_e is calculated using the following linear interpolation formula:

$$\mathbf{v}_e = (\mathbf{T}_e)^T \cdot [e_x \mathbf{T}_P \cdot \mathbf{v}_P + (1 - e_x) \mathbf{T}_N \cdot \mathbf{v}_N], \quad (49)$$

where e_x is the interpolation factor calculated as the ratio of geodesic distances \overline{eN} and \overline{PeN} (see Fig. 6):

$$e_x = \frac{\overline{eN}}{\overline{PeN}}, \quad (50)$$

and \mathbf{T}_P , \mathbf{T}_N and \mathbf{T}_e are the tensors of transformation from the global Cartesian coordinate system to the edge-based local orthogonal coordinate system, as defined in Fig. 6.

3.5. Calculation of surface tension

Regardless of the approach used to track the interface between the phases in a multiphase fluid flow, implementation of surface tension is always demanding. Unphysical ‘‘parasitic currents’’ will arise around the interface as a consequence of inaccuracy in the calculation of surface tension forces. Parasitic currents may be strong enough to destroy the interface and break the calculation.

From first principles, it follows that the total surface tension force on a closed surface must be identically equal to zero. This condition has served as starting point for a derivation of a novel procedure for calculation of surface tension forces.

Let us assume a phase interface is discretised with an unstructured surface mesh consisting of arbitrary polygonal control areas. Surface tension force acting on the control area S_{Af} (see Fig. 5) can be expressed by:

$$\mathbf{F}_{Af}^\sigma = \oint_{\partial S_{Af}} \mathbf{m} \sigma \, dL = \sum_e \int_{L_e} \mathbf{m} \sigma \, dL = \sum_e (\sigma \mathbf{m})_e L_e, \quad (51)$$

where L_e is the length of the edge e and $(\sigma \mathbf{m})_e$ is the surface tension force per unit length of the edge.

If the total surface tension force for each control area in the surface mesh is calculated using Eq. (51), then the total surface tension force for a closed surface will be exactly zero if vectors $(\sigma \mathbf{m})_e$ for two control areas sharing the edge e are parallel and have same magnitude and opposite direction.

It remains to decompose the surface tension force \mathbf{F}_{Af}^σ into the tangential component $(\nabla_s \sigma)_{Af}$ used in Eq. (30), and the normal component $(\kappa \sigma)_{Af} \mathbf{n}_{Af}$ used in Eq. (35). Using the surface Gauss' integral theorem (45), surface tension force acting on the control area S_{Af} can be expressed by the following equation:

$$\mathbf{F}_{Af}^\sigma = \int_{S_{Af}} \nabla_s \sigma \, dS + \int_{S_{Af}} \kappa \sigma \mathbf{n} \, dS. \quad (52)$$

When the right hand side of Eq. (52) is discretised using the mid-point rule and the result of discretisation is equalised with the right hand side of Eq. (51), the following expression is obtained:

$$(\nabla_s \sigma)_{Af} + (\kappa \sigma)_{Af} \mathbf{n}_{Af} = \frac{1}{S_{Af}} \sum_e (\sigma \mathbf{m})_e L_e. \quad (53)$$

Hence, the tangential component of the surface tension force acting on the control area S_{Af} is equal to the tangential component of the right hand side of Eq. (53):

$$(\nabla_s \sigma)_{Af} = \frac{1}{S_{Af}} (1 - \mathbf{n}_{Af} \mathbf{n}_{Af}) \cdot \sum_e (\sigma \mathbf{m})_e L_e, \quad (54)$$

and its normal component is equal to the respective normal component:

$$(\kappa \sigma)_{Af} \mathbf{n}_{Af} = \frac{1}{S_{Af}} (\mathbf{n}_{Af} \mathbf{n}_{Af}) \cdot \sum_e (\sigma \mathbf{m})_e L_e. \quad (55)$$

If the surface tension coefficient is constant ($\sigma = \text{const.}$), Eq. (54) will give zero tangential component of the surface tension force only if the normal unit vector of the control area S_{Af} satisfies the following equation:

$$\mathbf{n}_{Af} = \frac{\sum_e \sigma_e \mathbf{m}_e L_e}{|\sum_e \mathbf{m}_e L_e|}. \quad (56)$$

In the current implementation, the normal of a polygonal face is calculated by averaging normals of triangles obtained by decomposition of the face. Accordingly, in order to satisfy the zero total surface tension force condition for the closed surface, the tangential component of surface tension force defined by Eq. (54) is taken into account even in the case of constant surface tension coefficient.

With Eqs. (54) and (55) we shall formulate a procedure for calculation of surface tension force which ensures that the total surface tension force on the closed surface will be exactly zero. Unfortunately, fulfilment of this condition is not in itself sufficient for successful application of surface tension forces in the calculation. Specifically, unphysical fluid flow near the interface arises due to local (rather than global) inaccuracy in the calculation of surface tension forces.

One can see from Eq. (51) that the accuracy of the surface tension force calculation depends on the term $(\sigma \mathbf{m})_e$ which represents the surface tension force per unit length of the control area edge e . We calculated this term using the expression which follows from the approximation of the corresponding line integral over the edge e using the trapezoidal rule:

$$(\sigma \mathbf{m})_e = \frac{\sigma_i \hat{\mathbf{e}} \times \mathbf{n}_i + \sigma_j \hat{\mathbf{e}} \times \mathbf{n}_j}{2}, \quad (57)$$

where σ_i and σ_j are surface tension coefficients at points i and j . Use of the trapezoidal rule instead of the mid-point rule for approximation of the line integrals in Eq. (51) resulted in a significantly reduced amplitude of surface tension force oscillation along the interface.

From Eq. (57) one can see that the accuracy of the surface tension force calculation depends on the accuracy of calculation of unit normal vectors at the control area vertices. Vertex normals are calculated using least squares biquadratic surface fitting in the local coordinate system. Origin of the local coordinate system coincides with the vertex i for which the normal vector is calculated and its z -axis is parallel with the approximation of the normal vector at the vertex i calculated as the average normal of all faces belonging to the considered vertex i . The biquadratic surface

$$z(x, y) = ax^2 + by^2 + cxy + dx + ey, \quad (58)$$

passing through the vertex i in the local coordinate system is fitted through the set of vertices belonging to all faces sharing the considered vertex i .

Finally, mean curvature κ_{Af} in Eqs. (46) and (47) is calculated from Eq. (55) by assuming constant surface tension coefficient ($\sigma = \text{const.}$).

3.6. Solution procedure

The solution procedure is based on the well known PISO procedure [27] and consists of the following steps:

1. For the new time instance $t = t^n$, initialise the values of velocity, pressure and mass fluxes with the corresponding values from the previous time instance. The initial net mass fluxes through the faces at the interface are calculated with the corresponding face volume fluxes obtained using positions of the interface mesh points from the previous time step.

2. Define displacement directions for the interfacial mesh points and the control points.
3. Start of the outer iteration loop:
 - (a) In order to compensate the net mass flux through the interface, calculate displacement of the interface mesh points using procedure defined in Section 3.3.
 - (b) Displacement of the interface mesh points is used as a boundary condition for the solution of the mesh motion problem. After mesh movement, the new face volume fluxes are calculated using Eq. (28).
 - (c) Update pressure and velocity boundary conditions at the interface.
 - (d) Assemble and solve the discretised momentum Eq. (8) on the mesh with the current shape of the interface. The pressure field and face mass fluxes are used from the previous (outer) iteration.
 - (e) Start of the PISO iteration loop:
 - (i) Assemble and solve the discretised pressure Eq. (18) using velocity field obtained in the previous step.
 - (ii) Calculate new absolute mass fluxes through the cell faces using Eq. (19) with the new pressure field.
 - (iii) Calculate new cell-centre velocity using Eq. (15) with new pressure field.
 - (iv) If specified number of PISO iterations (usually 2) is not reached return to step (i).
 - (f) Calculate the net mass fluxes through the faces at the interface.
 - (g) Convergence is checked and if the residual levels and the net mass flux through the interface do not satisfy the prescribed accuracy, the procedure returns to step (a).
4. If the final time instance is not reached, return to step 1.

The efficiency of the above described solution procedure can be substantially increased if only interfacial mesh points are moved in step (b) instead of moving all mesh points. In that case, the entire mesh is deformed before the start of the outer iteration loop, according to the total interface displacement obtained in the previous time step. Thus, the mesh motion procedure is performed only once per time step. This approach is possible in cases where the magnitude of interface displacement in one time step is smaller than the thickness of the cells near the interface.

The time step size is limited by the Courant number criteria due to the PISO algorithm and temporal accuracy. However, in cases with a high surface tension force, a more stringent limitation emerges, due to explicit treatment of the surface tension force. The time step size must be sufficiently smaller than the period of the shortest capillary waves [46,25]:

$$\Delta t < \sqrt{\frac{\rho \min(L_{peN})}{2\pi\sigma}}, \quad (59)$$

where L_{peN} is the geodetic distance \overline{peN} (see Fig. 6).

It should be noted that the solution procedure described above can be modified in a straight-forward manner for a moving boundary problem where the motion of the boundary is prescribed in advance.

3.7. Mass conservation and Dirichlet fluid phase boundary condition

The mass conservation condition of the considered incompressible fluid phases depends on the satisfaction of the zero net mass flux condition at the interface. Experience shows that if adequate mesh resolution is used along the interface, the net mass flux through the interface can be reduced to a satisfactory level by performing sufficient number of outer iterations.

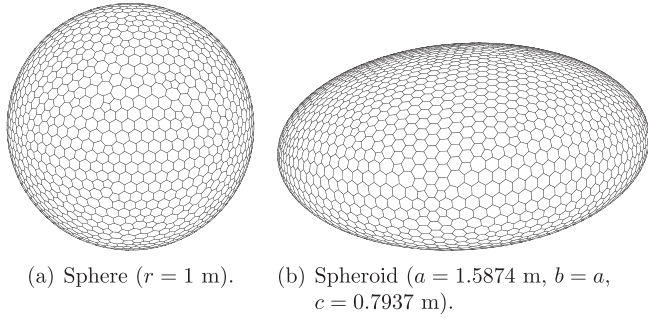


Fig. 7. Polygonal unstructured surface mesh for sphere and oblate spheroid used for evaluation of accuracy of curvature calculation.

When the incompressible fluid phase B is entirely enclosed by boundaries with prescribed velocity (Dirichlet boundary), gradual reduction of the interface net mass flux during iterations can cause problems with the solution of the pressure equation. Specifically, on the side B of the interface belonging to phase B , the mass flux is specified to ensure the zero net mass flux through the side B of the interface, *i.e.*

$$\dot{m}_{Bf}^n = \rho_{Bf} \dot{V}_{Bf}^n. \quad (60)$$

Here, \dot{V}_{Bf} is the volume flux of the face Bf at the side B of the interface. Term $\mathbf{n}_{Bf}^n \cdot (H/a)_f$ in pressure Eq. (18) is calculated for the faces at the side B of the interface using

$$\mathbf{n}_{Bf}^n \cdot \left(\frac{H}{a}\right)_{Bf} = \frac{\dot{m}_{Bf}^n}{\rho_{Bf}^n \mathbf{c}_{Bf}^n} + \left(\frac{1}{a}\right)_{Bf} \mathbf{n}_{Bf}^n \cdot (\nabla p)_{Bf}^n, \quad (61)$$

where the normal derivative of the dynamic pressure $\mathbf{n}_{Bf}^n \cdot (\nabla p)_{Bf}^n$ is specified using Eq. (34). If the phase B is incompressible, in order to solve the pressure equation, the pressure level must be fixed in one cell and the total mass flux through the side B of the interface $\sum_{Bf} \dot{m}_{Bf}$ must be identically equal to zero. While this total mass flux converges to zero during outer interactions, it will never be exactly zero. Therefore, the corrected mass flux through the side B of the interface is used in the pressure Eq. (18) as follows:

$$\dot{m}_{Bf}^c = \dot{m}_{Bf}^p - w_{Bf} \sum_{Bf} \dot{m}_{Bf}^p, \quad (62)$$

where \dot{m}_{Bf}^p is the mass flux through the face Bf obtained in the previous outer iteration and w_{Bf} is the weighting factor defined to ensure that the mass flux correction distribution is proportional to the mass fluxes distribution at side B ,

$$w_{Bf} = \frac{|\dot{m}_{Bf}^p|}{\sum_{Bf} |\dot{m}_{Bf}^p|}. \quad (63)$$

As the outer iterations advance, mass flux correction converges to zero.

4. Numerical results

Performance of the presented numerical methodology is tested on several test cases where numerical results are compared with available theoretical and experimental results. Validation of surface tension force calculation is carried out in the first test case while other cases validate the method on free-surface/interface flows.

4.1. Curvature of a sphere and oblate spheroid surfaces

Accuracy of surface tension force calculation in case of constant surface tension coefficient can be tested by evaluating accuracy of

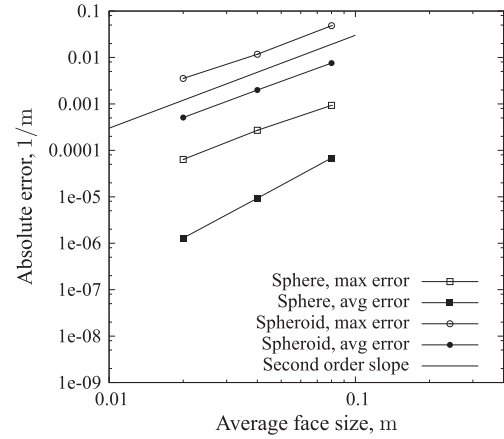


Fig. 8. Absolute error of calculated curvature for surfaces of sphere and oblate spheroid.

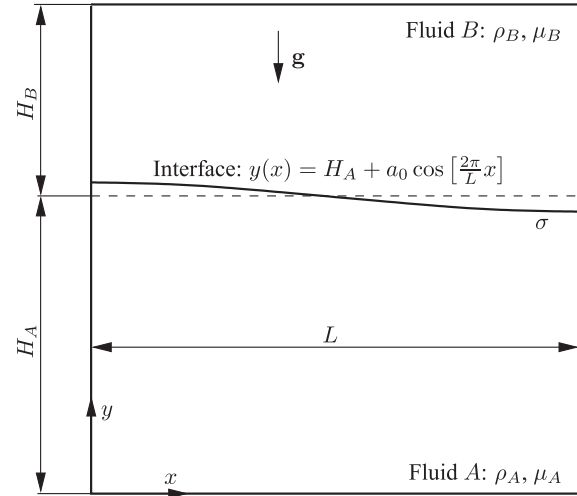


Fig. 9. Definition of spatial domain for small amplitude sloshing in 2-D rectangular tank.

curvature calculated using Eq. (55). Curvature is calculated for a sphere of unit radius and an oblate spheroid with following semi-axis lengths: $a = 1.5874$ m, $b = a$, $c = 0.7937$ m. Surfaces of sphere and spheroid are discretised by a polygonal surface mesh shown in Fig. 7.

Fig. 8 shows the absolute error of calculated curvature at face centres for different mesh resolutions. Both average and maximum error are reduced with second order rate. The difference between the maximum and average error represents the amplitude of the numerical noise of the calculated curvature which is the main cause of parasitic currents around the interface. According to the results shown in Fig. 8, the difference between the maximum and average error is also consistently reduced, which is favourable for stability of the interface tracking procedure. In [25], curvature is calculated directly from a biquadratic surface fit and it is noted that the calculated curvature slightly overpredicts the curvature of the sphere. This behaviour is not present in our approach.

4.2. Small amplitude sloshing in 2-D tank

Small amplitude sloshing in a 2-D rectangular tank is a common test case for unsteady fluid flow with a sharp interface/free-surface as an analytical solution obtained by linear theory is available. Spatial computational domain and initial shape of the interface

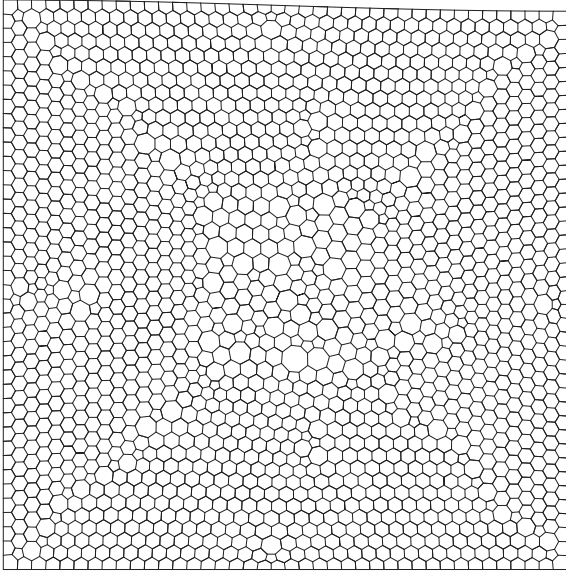


Fig. 10. Initial polygonal mesh over the bottom fluid phase domain for small amplitude sloshing in 2-D rectangular tank.

are defined in Fig. 9. Dimensions of the domain used in this study are: $L = 1$ m, $H_A = 1$ m, $H_B = 1$ m, $a_0 = 0.01$ m. Spatial discretisation (mesh) for the bottom fluid phase domain is shown in Fig. 10. The mesh is a uniform 2-D unstructured mesh consisting of arbitrary polygonal cells⁵ with 40 uniform segments along boundary edges. The whole outer boundary of the tank is treated as a slip wall.

In the first step, sloshing of one inviscid fluid phase due to combined action of gravity and surface tension force is analysed. In this case, the wave amplitude must remain constant in time and the natural frequency is defined by the following expression [47]:

$$f_0 = \sqrt{\left[\frac{\rho_A - \rho_B}{\rho_A + \rho_B} |\mathbf{g}| k + \frac{\sigma}{\rho_A + \rho_B} k^3 \right] \frac{\tanh(kH_A)}{4\pi^2}} \quad (64)$$

where $k = \pi/L$ is the wave number and \mathbf{g} is the gravity vector. Simulation is carried out for the following fluid properties: $\rho_A = 1$ kg/m³, $\mu_A = 0$ Pa s, $\sigma = 0.1$ N/m and gravitational acceleration vector is $\mathbf{g} = (0, -1, 0)$ m/s².

Fig. 11 shows the calculated position of the free-surface at the left side wall as a function of time for implicit Euler and backward temporal discretisation schemes. The time step size is constrained by Eq. (59) and it is the same in both cases. One can see that for the first order implicit Euler scheme wave amplitude decays significantly with time, while in the case of the second order accurate backward scheme wave amplitude remains constant with time. With the backward scheme, the influence of higher modes of oscillation is noticed, and amplitude and frequency of the extracted first mode are 0.009924 m and 0.39654 Hz, respectively, with the corresponding errors in respect to the analytical results being 0.7% and 0.09 %.

In the next step, sloshing of two viscous fluid phases due to the combined action of gravity and surface tension force is analysed. Simulation is performed for the following fluid properties: $\rho_A = 1$ kg/m³, $\rho_B = 0.01$ kg/m³, $\mu_A = 0.01$ Pa s, $\mu_B = 0.0001$ Pa s, $\sigma = 0.1$ N/m. The gravitational acceleration vector is $\mathbf{g} = (0, -1, 0)$ m/s². For this case, where kinematic viscosities of the two fluid phases are equal ($\nu_A = \nu_B$), Prosperetti [47] has derived the analytical expression which determines the time evolution of wave amplitude.

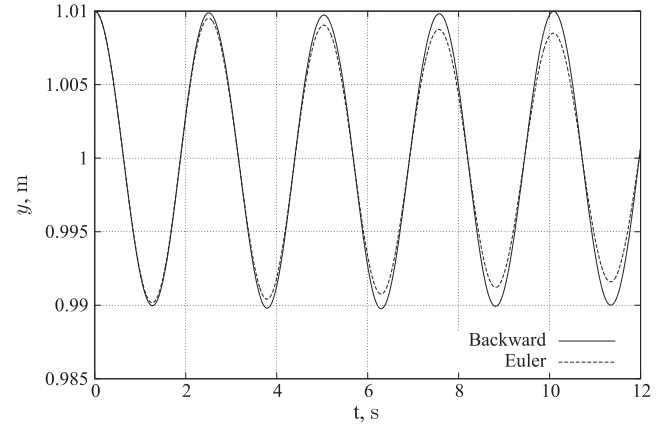


Fig. 11. Free-surface position at the left wall as a function of time for inviscid 2-D sloshing case.

Structure and resolution of the meshes for the top and bottom phases are equivalent to those shown in Fig. 10 and temporal discretisation is performed by using the backward scheme with the time step size constrained by Eq. (59). Fig. 12 shows the pressure field and velocity vectors near the interface at the time instance $t = 1$ s. Position of the interface at the left wall of the tank as a function of time is shown in the Fig. 13. One can see that the numerical solution is in excellent agreement with the corresponding analytical solution provided by Prosperetti [47].

In order to evaluate temporal accuracy of the method, the calculation is carried out for three different time step sizes: 0.004 s, 0.002 s and 0.001 s, with the error of the interface position at the left wall being calculated at the time instance $t = 1$ s with respect to the reference solution obtained with the time step size 0.0001 s. Results of the temporal accuracy analysis are shown in Fig. 14 for Euler and backward temporal discretisation schemes with the original Rhie–Chow (RC) interpolation and modified Rhie–Chow (mRC) interpolation. The Euler scheme gives the first order convergence rate of the error with both original and modified Rhie–Chow interpolation. Backward scheme gives the expected second order convergence rate of the error with modified Rhie–Chow interpolation while the original Rhie–Chow interpolation results in inconsistent behaviour.

4.3. Free oscillation of a water droplet

In this test case we shall consider free oscillation of an isolated water droplet with initial shape of a prolate ellipsoid. The droplet

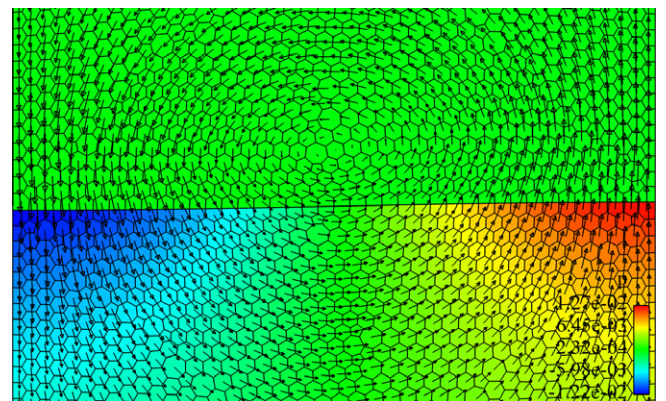


Fig. 12. Dynamic pressure field and velocity vectors around the interface for the viscous 2-D sloshing case.

⁵ In reality, the mesh is 3-D and consists of one layer of prismatic cells with arbitrary polygonal base.

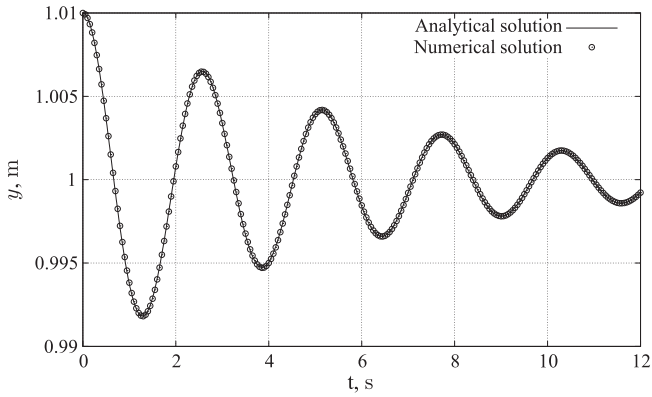


Fig. 13. Interface position at the left wall as a function of time for small amplitude sloshing of two viscous fluid phases in a 2-D rectangular tank due to gravity and surface tension force.

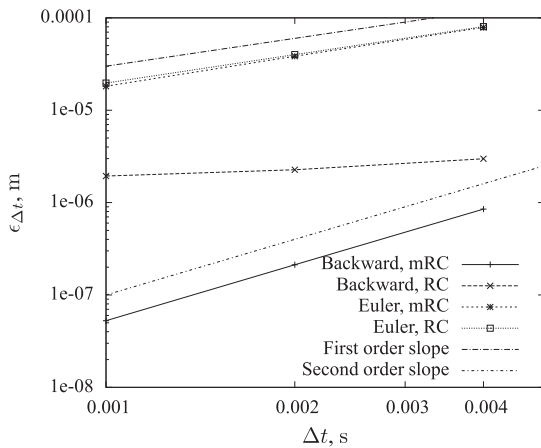


Fig. 14. Error of the interface position at the left wall at the time instance $t = 1$ s as a function of time step size for small amplitude sloshing of two viscous fluid phases in a 2-D rectangular tank due to gravity and surface tension force.

experiences a damped oscillation with surface tension as the only restoring force. Length of the major semi-axis of the ellipsoid is $a_0 = 0.00105$ m and the radius of a sphere with equivalent volume is $r_0 = 0.001$ m. Properties of the fluid used in the simulation are: $\mu_A = 0.001$ Pa s, $\rho_A = 998$ kg/m³, $\sigma = 0.073$ N/m. The numerical solution is compared with the Lamb's analytic solution. According to Lamb [48] the oscillation period is

$$T = \frac{2\pi}{\sqrt{\frac{n(n+1)(n-1)(n+2)}{(n+1)\rho_1} \frac{\sigma}{r_0^3}}}, \quad (65)$$

and the decay factor is

$$\tau = \frac{\rho_1 r_0^2}{(n-1)(2n+1)\mu_1}, \quad (66)$$

where n is the oscillation mode number with $n = 2$ for the lowest mode.

Calculation is performed on 8 parallel processors using the domain decomposition method and the MPI message passing interface standard [49]. The complete mesh consists of 62,213 polyhedral cells where the average size of polygonal faces at the droplet surface is 5×10^{-5} m, and the thickness of the first layer of cells near the free-surface amounts to 1×10^{-5} m. Fig. 15 shows a mesh partition used on a single processor. Presented numerical results are obtained using the backward temporal discretisation scheme.

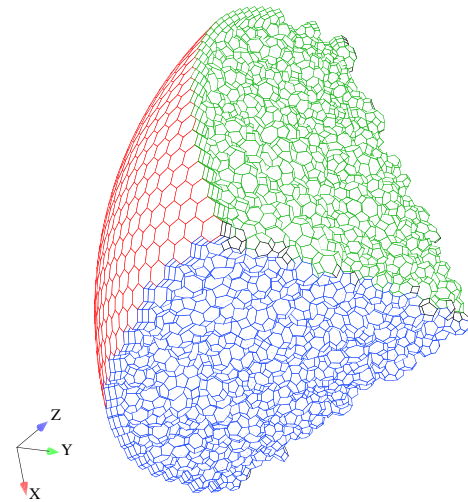


Fig. 15. Initial unstructured polyhedral mesh for droplet oscillation case (only a mesh partition used on a single processor is shown).

Fig. 16 shows the droplet shape and velocity vectors in the x - y plane cross-section at the time instance $t = 0.002$ s. Calculated lengths of the major and minor semi-axes of the droplet as a function of time are shown in Fig. 17. One can see that predicted decay of axes amplitude is in good agreement with the analytic solution. The calculated oscillation period is 0.00823 s, with the error of 0.195% compared to the theoretical oscillation period defined by Eq. (65). Temporal accuracy of the method is tested using the procedure described in the previous test case. The error of the major semi-axis length is evaluated at time $t = 0.002$ s. Results of the temporal accuracy validation are shown in Fig. 18, proving the second order temporal accuracy of the method. Finally, Fig. 19 shows the relative error of the droplet volume (ϵ_V) and the relative error of the droplet centre position (ϵ_C) as a function of time. It is clear that the numerical method causes minimal volume (mass) conservation error. Negligible displacement of the droplet centre from the initial position during simulation proves that the calculated surface tension force satisfies the zero total surface tension force condition for closed surfaces.

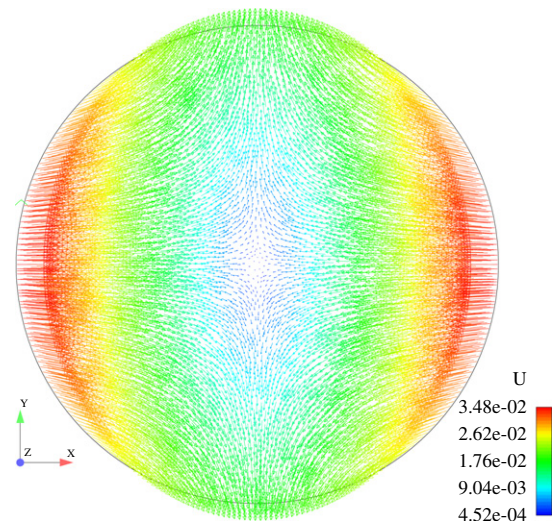


Fig. 16. Droplet shape and velocity vectors in x - y cross-section of the droplet at the time instance $t = 0.002$ s.

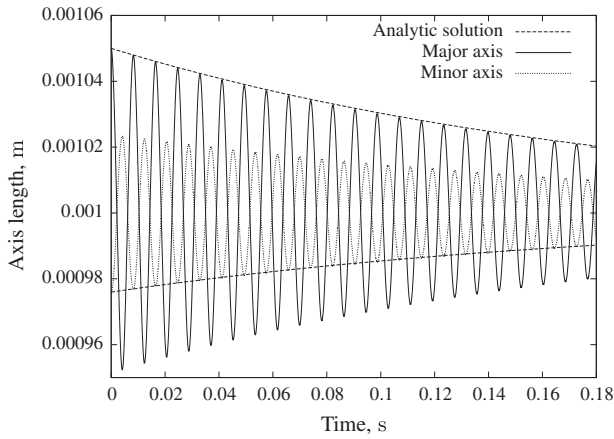


Fig. 17. Calculated droplet major and minor axis length as a function of time.

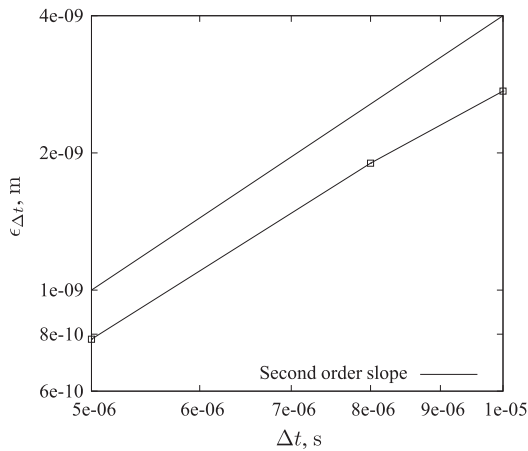


Fig. 18. Error of the calculated droplet axis length as a function of time step size.

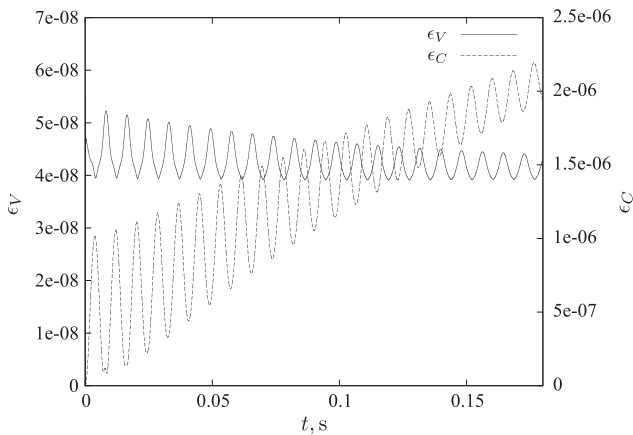


Fig. 19. Relative errors of droplet volume and droplet centre position as a function of time. Droplet volume error is relative in respect to initial volume and droplet position error is relative in respect to initial droplet equivalent radius.

4.4. Free rising air bubble in water

In the final case, we shall illustrate the accuracy of our method by simulating buoyant rise of a single air bubble in an unbounded space filled with pure quiescent water. We consider bubbles in the range of equivalent radius $r_b = (0.7\text{--}1.1)$ mm corresponding to the

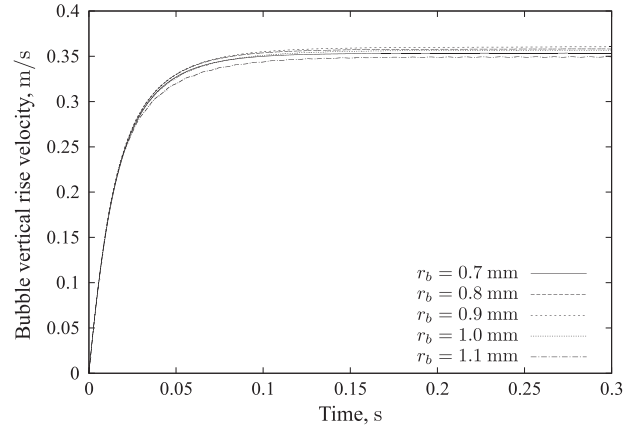


Fig. 20. Bubble vertical rise velocity as a function of time for different bubble equivalent radii.

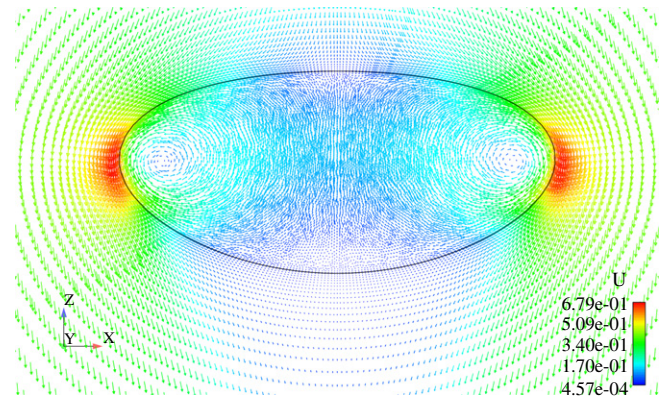


Fig. 21. Relative velocity vectors and final shape of the bubble in the central $x\text{--}z$ plane cross-section at the time instance $t = 0.3$ s for the bubble of equivalent radius $r_b = 1$ mm.

flow regime where surface tension forces have dominant influence as described by Tomiyama [50]. Smaller bubbles ($r_b < 0.91$) have a stable rectilinear path while larger bubbles have unstable zig-zag and/or helicoidal path [51,52]. The bubble shape in the considered range is stable and very closely resembles an oblate ellipsoid with the minor axis nearly in the direction of the tangent to the bubble path [53]. The wake behind the bubble with unstable path consists of two threads carrying vorticities of opposite sign [52].

Computational domain consists of the bubble volume and the volume of surrounding liquid bounded by the surface of a sphere of radius $20r_b$. In order to keep the bubble position in the centre of the spatial computational domain during the simulation, the calculation is performed in a moving non-inertial coordinate system. To switch from a fixed to the moving coordinate system, the acceleration of the moving coordinate system is added to the left hand side of the momentum Eq. (2) and the velocity at the inlet part of the outer boundary of the spatial domain is set to the negative value of the moving coordinate system velocity. Velocity and acceleration of the moving coordinate system are calculated using the procedure proposed by Rusche [54] in such a way to keep the bubble centre at the origin of the moving coordinate system.

Calculation is performed on 8 parallel processors as in the droplet oscillation case. For the 1 mm bubble, the air part of the mesh is the same as in the droplet case (see Fig. 15) with the exception of the spherical initial shape. The water part of the mesh consists of 287,900 arbitrary polyhedral cells (prisms with arbitrary polygonal base). The polygonal surface mesh at the water side of the interface

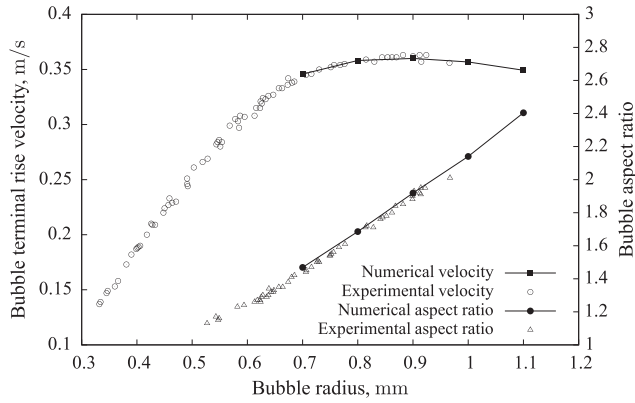


Fig. 22. Terminal vertical rise velocity and aspect ratio of the bubble as a function of the bubble equivalent radius. Numerical results are compared with experimental results obtained by Duineveld [51].

matches the one at the air side. The thickness of the first layer of cells near the interface is the same at both sides of the interface. The mesh for other bubbles is created by scaling the 1 mm bubble mesh. Mesh independence of the results obtained with the above described mesh for 1 mm bubble is verified by running the simulation on a mesh with increased resolution (approximately eight times more cells). The bubble rise velocity and the aspect ratio changed no more than 0.5%.

The simulation is performed with physical properties of water and air taken at 20 °C and 1 bar: $\mu_A = 0.001$ Pa s and $\rho_A = 998$ kg/m³ for water and $\mu_B = 1.8 \times 10^{-5}$ Pa s and $\rho_B = 1.25$ kg/m³ for air. Surface tension coefficient is $\sigma = 0.073$ N/m and gravitational acceleration vector is $\mathbf{g} = (0, 0, -9.81)$ m/s². The bubble, released as a sphere with zero velocity, starts to accelerate and after some time takes its final shape and reaches its terminal rise velocity. Fig. 20 shows the calculated vertical rise velocity as a function of time for various bubbles where one can see that the terminal rise velocity is reached approximately after 0.15 s which is in agreement with the experimental observations reported in [55]. For the final shape of the 1 mm bubble, Fig. 21 shows the relative velocity vectors in the *x*–*z* plane cross-section inside and outside the bubble.

Numerical results obtained using our method are compared with experimental results obtained by Duineveld [51], who measured the velocity and the shape (aspect ratio) of the rising bubbles, with an equivalent radius of $r_b = (0.33\text{--}1.00)$ mm, in “hyper

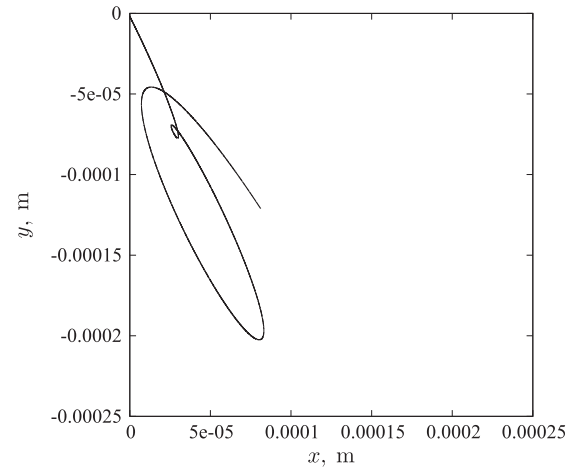


Fig. 24. Projection of bubble path onto a horizontal plane for 1 mm bubble obtained using increased mesh resolution.

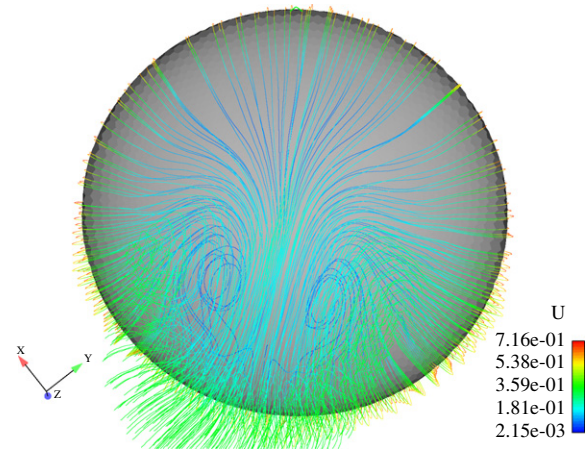


Fig. 25. Stream lines behind the bubble of equivalent radius $r_b = 1$ mm, showing formation of double tread vortices.

clean” water at 20 °C using a very gentle bubble injection method. Fig. 22 shows the comparison between numerical and experimental results for the bubble rise velocity and aspect ratio with excellent agreement.

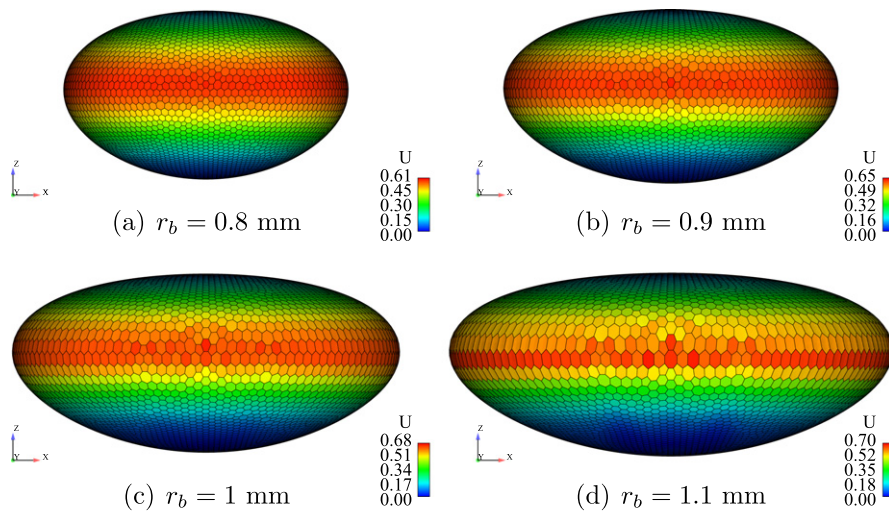


Fig. 23. Interface coloured by relative velocity magnitude for different bubble equivalent radii.

Fig. 23 shows the bubble interfaces for different bubble sizes. The interfaces are represented by the 3-D surface polygonal mesh coloured by the relative velocity magnitude. The shape of the smallest bubble is close to an prolate ellipsoid. As the bubble size increases, the fore-aft symmetry of the bubbles is lost which is in agreement with the observations by Duineveld [51].

Numerical results show that the bubbles in the considered range of equivalent radius rise along the stable rectilinear paths with the stable terminal rise velocity (see Fig. 20). Very slow development of path instability is noticed for the 1.1 mm bubble. Mesh independence analysis showed that postponed path instability is a consequence of insufficient mesh resolution around the bubble. Although the terminal rise velocity and the aspect ratio of the 1 mm bubble did not change more than 0.5% by increasing mesh resolution, it is noticed that the increased mesh resolution was necessary in order to develop path instability. Fig. 24 shows the projection of the 1 mm bubble path onto a horizontal plane during the transition from rectilinear to zig-zag obtained with the increased mesh resolution. The double tread wake developed behind the bubble can be seen in Fig. 25.

5. Conclusions

A moving mesh interface tracking method implemented in OpenFOAM is described and evaluated. The method is primarily intended for simulating elementary 3-D two-phase interfacial fluid flows such as motion of isolated droplets and bubbles where strong surface tension forces exist.

The interface represented by a mesh boundary is tracked in a semi-implicit manner inside the iterative PISO procedure for the numerical solution of incompressible fluid flow on a moving mesh. Second order accuracy in space is accomplished by the collocated FV method with arbitrary polyhedral mesh support. The polyhedral mesh is adjusted to the time varying shape of the interface using a specially designed vertex-based FE mesh motion solver with polyhedral mesh support. In order to accomplish second order accuracy in time with backward temporal discretisation scheme, GCL is properly taken into account and original Rhie–Chow interpolation is modified to prevent time step size dependency. Surface tension forces are calculated on arbitrary polygonal surface meshes with second order accuracy using a novel “force-conservative” approach.

Evaluation of the method on the three presented test cases shows very good agreement with available theoretical and experimental results. Second order accuracy of the method in time is proved on the 2-D sloshing case and the 3-D droplet oscillation case. Simulation of buoyant rise of an air bubble in quiescent water showed that the method can capture all relevant features of the flow, from the shape and the terminal rise velocity of the bubble to the path oscillation and the double thread wake behind the bubble.

Arbitrary polyhedral unstructured mesh support of the method opens up the possibility for implementation of topological mesh changes necessary to simulate more complicated interfacial flows, such as merging and breaking of bubbles or droplets. Work in that direction is in progress and will be reported in future publications.

Presence of surface-active agents (surfactants) at the interface is a very common situation. Surfactants typically reduce the surface tension coefficient and their non-uniform distribution due to convective and diffusive transport along the interface causes the gradient of surface tension, known as the Marangoni stress, which acts on the fluid near the interface with a tendency to restore uniform surfactant concentration. Implementation of the FAM partly described in this paper allows us to discretise the surface transport equations on an arbitrary polygonal surface mesh. This methodology allows analysis of the influence of the surfactants on the interfacial fluid flows, and will be the subject of a follow-up paper.

References

- [1] Harlow FH, Welch JE. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys Fluids* 1965;8(12):2182–9.
- [2] McKee S, Tome MF, Ferreira VG, Castelo A, Sousa FS, Mangiacavacchi N. The MAC method. *Comput Fluids* 2008;37(8):907–30.
- [3] Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *J Comput Phys* 1981;39:201–25.
- [4] Gueyffier D, Li J, Scardovelli R, Zaleski S. Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *J Comput Phys* 1999;152:423–56.
- [5] Queutey P, Visonneau M. An interface capturing method for free-surface hydrodynamic flows. *Comput Fluids* 2007;36(9):1481–510.
- [6] Osher S, Fedkiw RP. Level set methods: an overview and some recent results. *J Comput Phys* 2001;169:463–502.
- [7] Losasso F, Fedkiw R, Osher S. Spatially adaptive techniques for level set methods and incompressible flow. *Comput Fluids* 2006;35(10):995–1010.
- [8] Fedkiw RP, Aslam T, Merriman B, Osher S. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *J Comput Phys* 1999;152:457–92.
- [9] Juric D, Tryggvason G. A front-tracking method for dendritic solidification. *J Comput Phys* 1996;123(1):127–48.
- [10] Tryggvason G, Bunner B, Esmaeili A, Juric D, Al-Rawahi N, Tauber W, et al. A front-tracking method for the computations of multiphase flow. *J Comput Phys* 2001;169(2):708–59.
- [11] Ryskin G, Leal LG. Numerical solution of free-boundary problems in fluid mechanics. Part 1: the finite-difference technique. *J Fluid Mech* 1984;148:1–17.
- [12] Ryskin G, Leal LG. Numerical solution of free-boundary problems in fluid mechanics. Part 2: buoyancy-driven motion of a gas bubble through a quiescent liquid. *J Fluid Mech* 1984;148:19–35.
- [13] Ryskin G, Leal LG. Numerical solution of free-boundary problems in fluid mechanics. Part 3: bubble deformation in an axisymmetric straining flow. *J Fluid Mech* 1984;148:37–43.
- [14] Takagi S, Matsumoto Y, Huang H. Numerical analysis of a single rising bubble using boundary-fitted coordinate system. *JSME Int J* 1997;40(1):42–50.
- [15] Yang B, Prosperetti A. A second-order boundary-fitted projection method for free-surface flow computations. *J Comput Phys* 2006;213:574–90.
- [16] Dai M, Schmidt DP. Adaptive tetrahedral meshing in free-surface flow. *J Comput Phys* 2005;208:228–52.
- [17] Quan S, Schmidt DP. A moving mesh interface tracking method for 3D incompressible two-phase flows. *J Comput Phys* 2007;221:761–80.
- [18] Quan S, Lou J, Schmidt DP. Modeling merging and breakup in the moving mesh interface tracking method for multiphase flow simulations. *J Comput Phys* 2009;228:2660–75.
- [19] Weller HG, Tabor G, Jasak H, Fureby C. A tensorial approach to computational continuum mechanics using object orientated techniques. *Comput Phys* 1998;12(6):620–31.
- [20] Demirdžić I, Perić M. Space conservation law in finite volume calculations of fluid flow. *Int J Numer Methods Fluids* 1988;8:1037–50.
- [21] Geuzaine P, Grandmont C, Farhat C. Design and analysis of ALE schemes with provable second-order time-accuracy for inviscid and viscous flow simulations. *J Comput Phys* 2003;191:206–27.
- [22] Farhat C, Geuzaine P. Design and analysis of robust ALE time-integrators for the solution of unsteady flow problems on moving grids. *Comput Methods Appl Mech Eng* 2004;193:4073–95.
- [23] Muzafherija S, Perić M. Computation of free-surface flows using the finite-volume method and moving grids. *Numer Heat Transfer B* 1997;32:369–84.
- [24] Tuković Ž. Finite volume method on domains of varying shape (in Croatian). Ph.D. thesis, Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb; 2005.
- [25] Dai M, Wang H, Perot JB, Schmidt DP. Direct interface tracking of droplet deformation. *Atomiz Sprays* 2002;12:721–35.
- [26] Perot B, Nallapati R. A moving unstructured staggered mesh method for the simulation of incompressible free-surface flows. *J Comput Phys* 2003;184:192–214.
- [27] Issa RI. Solution of the implicitly discretised fluid flow equations by operator-splitting. *J Comput Phys* 1986;62(1):40–65.
- [28] Rhie CM, Chow WL. A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation. *AIAA J* 1983;21:1525–32.
- [29] Ferziger JH, Perić M. Computational methods for fluid dynamics. Berlin-New York: Springer Verlag; 1995.
- [30] Choi SK. Note on the use of momentum interpolation method for unsteady flows. *Numer Heat Transfer A* 1999;36:545–50.
- [31] Shen WZ, Michelsen JA, Sørensen JN. Improved Rhie–Chow interpolation for unsteady flow computations. *AIAA J* 2001;39(12):2406–9.
- [32] Yu B, Kawaguchi Y, Tao W-Q, Ozoe H. Checkerboard pressure predictions due to the underrelaxation factor and time step size for a nonstaggered grid with momentum interpolation method. *Numer Heat Transfer B* 2002;41:85–94.
- [33] Yu B, Tao W-Q, Wei J-J, Kawaguchi Y, Tagawa T, Ozoe H. Discussion on momentum interpolation method for collocated grids of incompressible flow. *Numer Heat Transfer B* 2002;42:141–66.
- [34] Thomas PD, Lombard CK. Geometric conservation law and its application to flow computations on moving grids. *AIAA J* 1979;17:1030–7.

- [35] Batchelor FR. An introduction to fluid dynamics. Cambridge: Cambridge University Press; 1967.
- [36] Chen KP, Saric W, Stone HA. On the deviatoric normal stress on a slip surface. *Phys Fluids A* 2000;12(12):3280–1.
- [37] Jasak H, Tuković Ž. Automatic mesh motion for the unstructured finite volume method. *Trans FAMENA* 2006;30(2):1–20.
- [38] Zienkiewicz OC. The finite element method in engineering science. London: McGraw Hill; 1971.
- [39] Jasak H, Weller HG, Gosman AD. High resolution NVD differencing scheme for arbitrarily unstructured meshes. *Int J Numer Methods Fluids* 1999;31:431–49.
- [40] Jasak H. Error analysis and estimation for finite volume method with applications to fluid flows. Ph.D. thesis, Imperial College, University of London; 1996.
- [41] Demirdžić I, Muzaferija S. Numerical method for coupled fluid flow, heat transfer and stress analysis using unstructured moving meshes with cells of arbitrary topology. *Comput Methods Appl Mech Eng* 1995;125:235–55.
- [42] Jasak H, Weller HG. Application of the finite volume method and unstructured meshes to linear elasticity. *Int J Numer Methods Eng* 2000;48:267–87.
- [43] Issa RI, Oliveira PJ. Numerical prediction of phase separation in two-phase flow through T-junction. *Comput Fluids* 1994;23(2):347–72.
- [44] Edwards DA, Brenner H, Wasan DT. Interfacial transport processes and rheology. Butterworth-Heinemann; 1991.
- [45] Weatherburn CE. Differential geometry of three dimensions. London: Cambridge University Press; 1972.
- [46] Brackbill JU, Kothe DB, Zemach C. A continuum method for modelling surface tension. *J Comput Phys* 1992;100:335–54.
- [47] Prosperetti A. Motion of two superposed viscous fluids. *Phys Fluids* 1981;24(7):1217–23.
- [48] Lamb H. Hydrodynamics. Cambridge University Press; 1975.
- [49] Gropp W, Lusk E, Doss N, Skjellum A. A high performance, portable implementation of the MPI message passing interface standard. *Parallel Comput* 1996;22:789–828.
- [50] Tomiyama A, Celata GP, Hosokawa S, Yoshida S. Terminal velocity of single bubbles in surface tension force dominant regime. *Int J Multiphase Flow* 2002;28:1497–519.
- [51] Duineveld PC. The rise velocity and shape of bubbles in pure water at high Reynolds number. *J Fluid Mech* 1995;292:325–32.
- [52] de Vries AWG, Biesheuvel A, van Wijngaarden L. Notes on the path and wake of a gas bubbles rising in pure water. *Int J Multiphase Flow* 2002;28:1823–35.
- [53] Ellingsen R, Risso F. On the rise of an ellipsoidal bubble in water: oscillatory paths and liquid-induced velocity. *J Fluid Mech* 2001;440:235–68.
- [54] Rusche H. Computational fluid dynamics of dispersed two-phase flows at high phase fractions. Ph.D. thesis, Imperial College, University of London; 2002.
- [55] Shew WL, Poncet S, Pinton J-F. Force measurements on rising bubbles. *J Fluid Mech* 2006;569:51–60.