INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2012; **69**:1496–1521
Published online 8 August 2011 in Wiley Online Library (wileyonlinelibrary.com/journal/nmf). DOI: 10.1002/fld.2648

Experimental and numerical analysis of a sphere falling into a viscous fluid

Marcela Cruchaga[1,*,†], Rainald Löhner[2] and Diego Celentano[3]

[1] *Departamento de Ingeniería Mecánica, Universidad de Santiago de Chile (USACH), Av. Bdo. O'Higgins 3363, Santiago, Chile*
[2] *Center for Computational Fluid Dynamics, Department of Computational and Data Sciences, M.S. 6A2, George Mason University, Fairfax, VA 22030-4444, USA*
[3] *Departamento de Ingeniería Mecánica y Metalúrgica, Pontificia Universidad Católica de Chile, Av. Vicuña Mackenna 4860, Santiago, Chile*

SUMMARY

The experimental and numerical analysis of spheres falling into viscous flows is considered. The physical model is built using a set of silicone and glass spheres falling into oil and water. The rigid-body trajectory of the sphere and the free surface evolution are obtained from videos. The numerical results are obtained using two different finite element codes. The first code uses a fractional step approach with adaptive meshes and time-step sizes whereas the second code uses a monolithic fully coupled fixed-mesh technique. The results exhibit a good comparison between both numerical techniques and with the experiments. Copyright © 2011 John Wiley & Sons, Ltd.

Received 6 December 2010; Revised 13 April 2011; Accepted 7 June 2011

KEY WORDS: viscous flows; numerical verification; experimental validation; finite element solutions

1. INTRODUCTION

Many engineering applications require the computation of flow problems that include the motion of fluid–solid and fluid–fluid interfaces. Examples are sediment flows commonly found in mining applications, fluidized beds to transport sand, gravel or even food, and polymer and melt applications. In all of these cases, the accurate description of the interfaces, which may exhibit considerable motion and/or topology change, presents major numerical challenges.

The modeling of the interaction of fluid flow with rigid or flexible solid structures can be performed using different approaches: arbitrary Lagrangian–Eulerian (ALE) strategies [1–7], deforming-spatial-domain/stabilized space–time (DSD/SST) formulations [8–10] where the fully discretized equations are solved with either iterative or direct coupling methods, particle FEM (PFEM) [11], immersed boundary (IB) techniques [12, 13], and moving Lagrangian interface techniques (MLIT) [14]. Focusing on the description of interfaces, we applied basically two methodologies. In the moving mesh techniques, like ALE and DSD/SST, the mesh moves to follow the interface, providing the fine mesh resolution needed near the fluid–solid interface [15–18]. Nevertheless, in the computation of very complex and unsteady fluid–fluid interfaces, the remeshing required in such techniques can be a very time-consuming task. To avoid this inconvenience, researchers have developed fixed-mesh techniques to describe complex moving interfaces with the understanding that the accuracy near the interface will be limited by the mesh resolution where the interface happens to be located at a given instant [13, 19–22].

[*] Correspondence to: Marcela Cruchaga, Departamento de Ingeniería Mecánica Universidad de Santiago de Chile (USACH) Av. Bdo. O'Higgins 3363, Santiago, Chile.
[†] E-mail: marcela.cruchaga@usach.cl

When a flow problem involves both fluid–solid and fluid–fluid interfaces, it is possible to define a mixed method, so that each type of interface is handled with the technique most appropriate for that type of interface. In the mixed interface-tracking/interface-capturing technique (MITICT), fluid–solid interfaces are accurately tracked with a moving mesh method (interface-tracking technique), and fluid–fluid interfaces, if they are too complex and unsteady to track, are treated with a fixed-mesh (interface-capturing) technique. Different MITICT type formulations can be found in the literature [16, 22–25].

Two numerical formulations are used for the present analysis, which considers the problem of a sphere falling into a liquid-filled tube, where the upper part of the tube is occupied by air and the lower part by a liquid (Figure 1). The sphere first falls through the air, then it impacts the liquid surface, and then it falls through the liquid. At the same time, experiments were conducted to verify the results obtained from the different numerical techniques.

The first, a 3D code that has been in continuous development for over a decade, implements the techniques presented and tested in previous studies [2, 12, 16, 20, 26–33]. Both moving body-fitted mesh options, as well as fixed-mesh options such as embedded surfaces [18, 33] and immersed bodies [12], can and have been used.

The second, a monolithic fully coupled code, implements the techniques recently presented and tested [22]. The fluid–solid interface motion is described with the moving Lagrangian interface technique (MLIT) [14] and the fluid–fluid interfaces with the edge-tracked interface locator technique (ETILT) [23, 34]. Both techniques are fixed-mesh FEMs. In this context, the equation of motion for a rigid particle is recovered by using a high-viscosity value in the solid region. Although it is a monolithic fully coupled 3D Navier–Stokes code, an axisymmetric model is used because of the low Reynolds number present in the experiments. We note that the physical tests do not exhibit 3D effects. In addition, the axisymmetric assumption reduces the computational time.

Numerical predictions and experiments of settling spheres have also been reported in other works (e.g., [14, 35]) using other materials. Nevertheless, the interface evolution is not reported. In particular, an exhaustive analysis of wall distance effects, boundary conditions, time-step size, terminal settling velocity, and mesh resolution needed near the fluid–solid interface has been presented in [14]. Other contributions related to this subject are the simulations of multiple spheres falling in a liquid-filled tube [36–39] and the experimental study of the hydrodynamics of spherical projectiles impacting on a free surface using high-speed imaging techniques [40].

The main objective of the present work is to report the results of experiments and their numerical simulation by using the codes previously referred to. In spite of the inherent experimental limitations of the physical model, the observation of the phenomena and the measurements help to evaluate and to validate the numerical predictions. Additionally, an effort is made to compare computational solutions using different formulations.

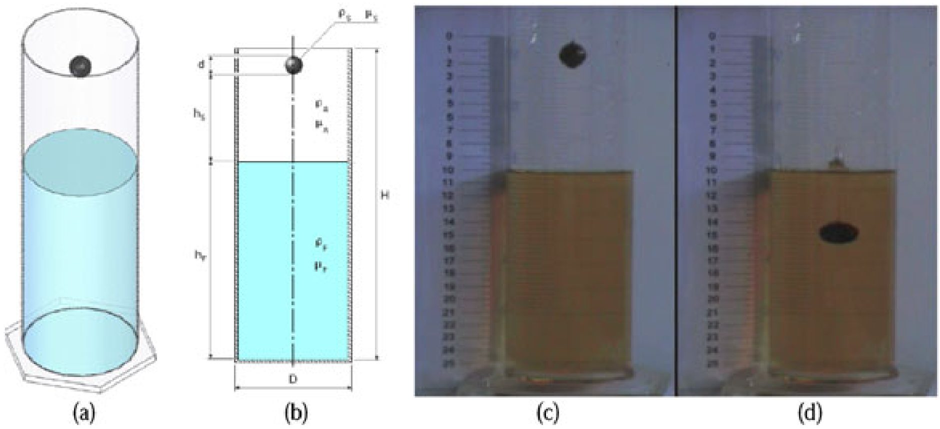


Figure 1. (a–d) Problem description and experimental setup.

DOI: 10.1002/fld

The remainder of the paper is organized as follows: Section 2 describes the experimental setup, as well as an error analysis of the results obtained. Section 3 describes the numerical procedures employed. This is followed in Section 4 by an error analysis for the numerical procedures that is intended to identify the grid and time-step requirements. Section 5 compares the experimental and numerical results obtained. Finally, some conclusions are drawn in Section 6, together with a suggestion for further investigations.

2. EXPERIMENTAL SETUP

An experimental campaign was conducted in order to validate numerical results. Figure 1 shows the experimental setup considered. A sphere falls down a tube that is partially filled with a liquid. Three different regimes may be identified in the gravity-driven sphere motion: first, a basically friction-free particle acceleration, then a deceleration under viscous and free surface effects after particle impingement on the liquid surface, and finally a viscous, steady-state deceleration until contact with the bottom wall occurs or, for those instances when the density of the sphere is smaller than the density of the liquid, a gradual flotation towards the final semi-submerged position. The motion of the sphere was recorded with a high-velocity video camera that had a frame rate of 30 frames per second. These frames were then post-processed in order to obtain the motion of the sphere and the free surface in time.

The tube has a diameter of $D = 80$ mm and a length of $H = 250$ mm, and it is filled with liquid up to $h_f = 150$ mm. The particular cases considered here are spheres of silicone and glass of equal diameter $d = 15$ mm. The silicone and glass densities are 962 and 3594 kg/m^3, respectively. The chosen liquids are oil and water with densities of 911 and 1000 kg/m^3, respectively. With these two fluids different settling, velocities are expected. Three cases are studied in detail here: silicone–oil, glass–oil, and silicone–water. In order to achieve repeatability, we carried out several tests, on the order of ten, for each case.

Figure 2 illustrates the processing of the frames obtained for the silicone sphere falling into oil case. The same methodology was used for the glass–oil and silicone–water cases. The experimental

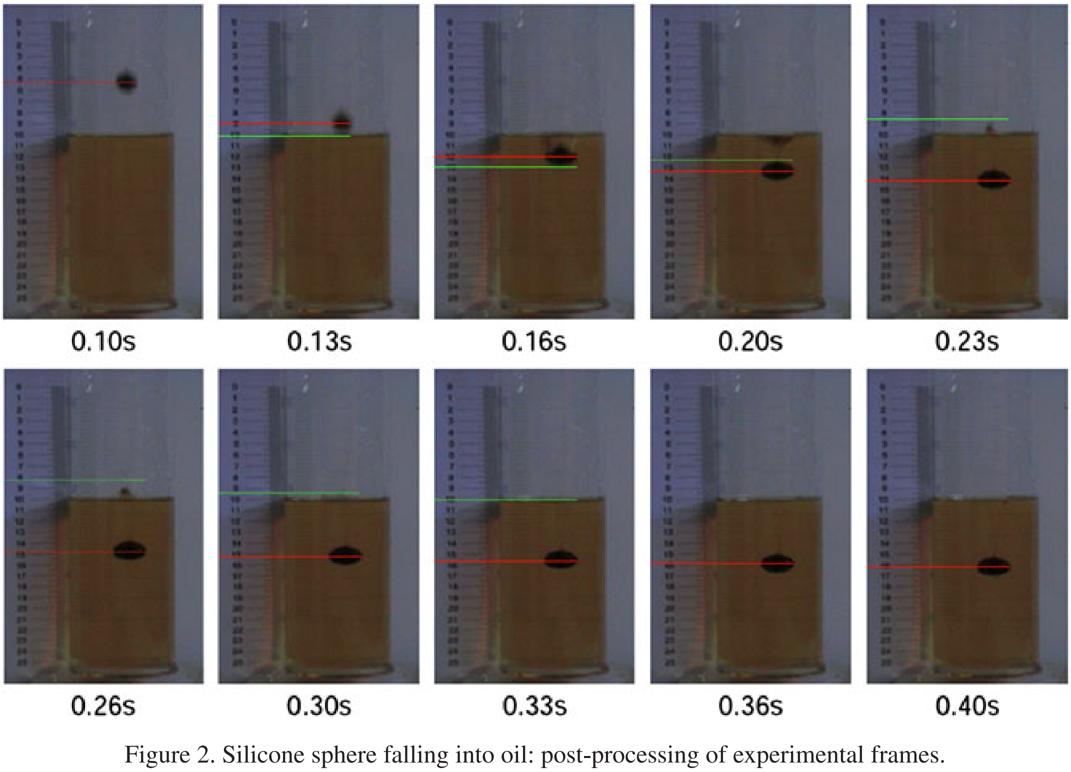


Figure 2. Silicone sphere falling into oil: post-processing of experimental frames.

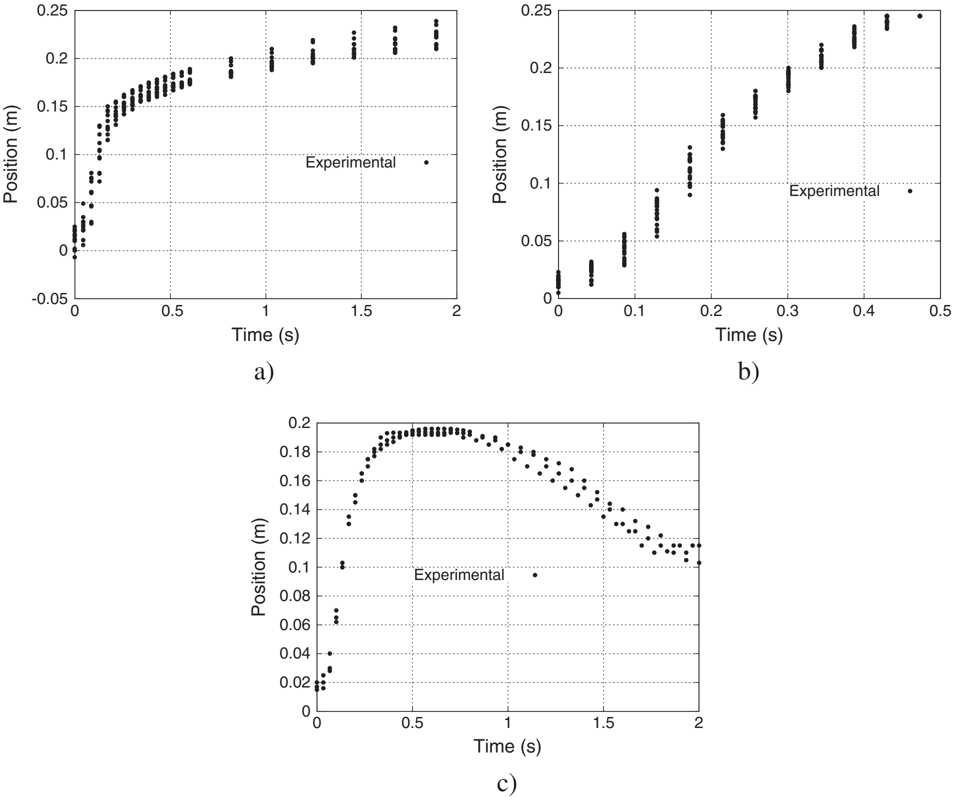
DOI: 10.1002/fld

measurements of the time evolution of the position (taken as the distance from its initial position) of the sphere are plotted in Figure 3 for the cases of silicone–oil, glass–oil, and silicone–water. In Figure 3, the different regimes expected can easily be identified. Nevertheless, the data captured during the gravity-driven friction-free particle acceleration are not enough to confirm the gravity law, that is, the error in time does not enable us to determine such a regime. In spite of this, the viscous motion is well captured. For the silicone–water case, the number of frames per second was not enough to track the freesurface. Moreover, the silicone–water experiments exhibit incipient 3D effects, for example, small horizontal displacements when the sphere reaches its lower vertical position. To illustrate such aspect, we plotted a set of shots for the silicone–water case in Figure 4.

Although there are clear limitations in the accuracy of the measurements, the experimental data help in the validation of the numerical results.

Let us consider the errors induced by the present experimental setup. Given that the frame rate and the release of the spheres could not be synchronized, the exact release time may have a maximum error of order 0.043 s. As the sphere travels approximately 8 cm before touching the liquid, this takes of the order of 0.127 s. The arrival time could therefore be in error by approximately ±25%. The initial position of the sphere also showed a variation of ±5 mm, that is, ±6.25% of the release height. Given that the arrival times and velocities are proportional to the square root of the release height, the errors in the quantities are approximately half of the error of the initial position, that is, in the order of ±3.1%.

The errors in the free surface height were estimated to be of the order of ±5 mm. During the viscous settling phase of the spheres inside the liquids, some experiments exhibited small air bubbles that attached themselves to the spheres upon entry. These air bubbles became clearly visible after they separated from the sphere upon impact on the bottom of the tube. Having an air bubble



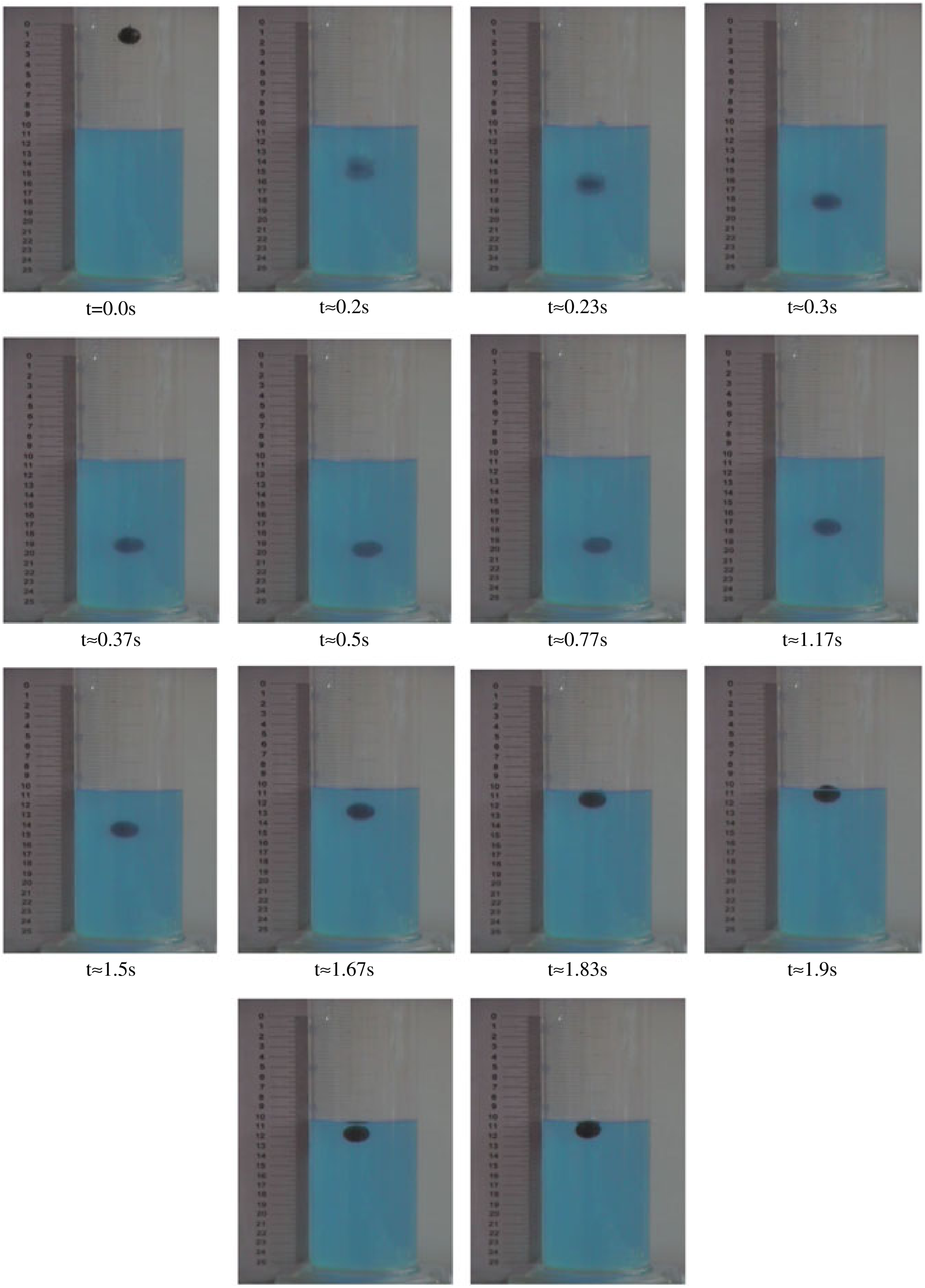
a)

b)

c)

Figure 3. Sphere falling into a liquid, history of the sphere position: (a) sphere of silicone into oil, (b) sphere of glass into oil, and (c) sphere of silicone into water.

DOI: 10.1002/fld



t=0.0s t≈0.2s t≈0.23s t≈0.3s

t≈0.37s t≈0.5s t≈0.77s t≈1.17s

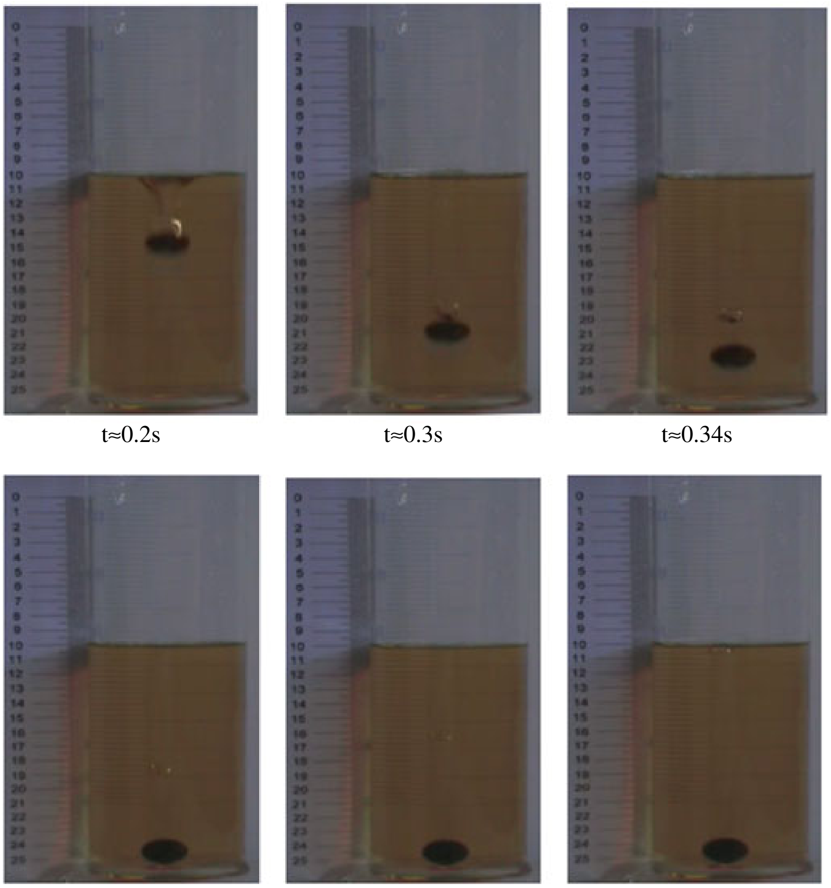
t≈1.5s t≈1.67s t≈1.83s t≈1.9s

t≈2.07s t≈2.40s

Figure 4. Silicone sphere falling into water.

attached to the sphere will decrease the final velocity due to an apparent reduction of density. This aspect was apparent in the glass–oil case as shown in Figure 5. The error in final velocity estimated for some of the cases was of the order of ±10%.

DOI: 10.1002/fld



t≈0.2s t≈0.3s t≈0.34s

t≈0.44s t≈0.57s t≈0.87s

Figure 5. Bubble formation in the case of a glass sphere falling into oil.

In summary, even though this is a seemingly very simple set of experiments, the error margins are of the order of 3–5%. It is therefore pointless to attempt to drive the numerical errors below this range.

3. GOVERNING EQUATIONS

The Navier–Stokes equations of unsteady incompressible flows are written as

$$\rho \mathbf{v}_t + \rho \mathbf{v} \nabla \mathbf{v} + \nabla p = \nabla \mu \epsilon + \rho \mathbf{g}, \tag{1}$$

$$\nabla \cdot \mathbf{v} = 0, \tag{2}$$

where ρ, $\mathbf{v}$, p, μ, ϵ, and $\mathbf{g}$ are the density, velocity, pressure, dynamic viscosity, strain-rate tensor, and gravity vector. This system of equations is complemented with a set of initial and boundary conditions. To describe the interface motion, we have to consider an additional equation. The two approaches considered in this work are described in Section 4.

4. DISCRETIZED FORMULATIONS

4.1. Body-fitted technique

This numerical formulation, which has been in development for over a decade, is based on the following criteria:

DOI: 10.1002/fld

- spatial discretization using unstructured grids (in order to allow for arbitrary geometries and adaptive refinement);
- spatial approximation of unknowns with simple finite elements (in order to have a simple input/output and code structure);
- temporal approximation using implicit integration of viscous terms and pressure (the interesting scales are the ones associated with advection);
- temporal approximation using explicit integration of advective terms;
- low-storage, iterative solvers for the resulting systems of equations (in order to solve large 3D problems); and
- steady results that are independent from the time step chosen (in order to have confidence in convergence studies).

4.1.1. Temporal discretization. For the applications considered here, the important physical phenomena propagate with the advective time scales. We will therefore assume that the advective terms require an explicit time integration. Diffusive phenomena typically occur at a much faster rate and can/should therefore be integrated implicitly. Given that the pressure establishes itself immediately through the pressure-Poisson equation, an implicit integration of pressure is also required. The hyperbolic character of the advection operator and the elliptic character of the pressure-Poisson equation have led to a number of so-called projection schemes. The key idea is to predict first a velocity field from the current flow variables without taking the divergence constraint into account. In a second step, the divergence constraint is enforced by solving a pressure-Poisson equation. The velocity increment can therefore be separated into an advective-diffusive and pressure increment:

$$\mathbf{v}^{n+1} = \mathbf{v}^n + \Delta\mathbf{v}^a + \Delta\mathbf{v}^p = \mathbf{v}^* + \Delta\mathbf{v}^p \ . \tag{3}$$

For an explicit (forward Euler) integration of the advective terms, with implicit integration of the viscous terms, one complete time step is given by the following:

- *Advective-diffusive prediction*: $\mathbf{v}^n \to \mathbf{v}^*$

$$\left[\frac{\rho}{\Delta t} - \theta\nabla\mu\nabla\right]\left(\mathbf{v}^* - \mathbf{v}^n\right) + \mathbf{v}_a^n \cdot \nabla\mathbf{v}^n + \nabla p^n = \nabla\mu\nabla\mathbf{v}^n + \rho\mathbf{g} \ ; \tag{4}$$

- *Pressure correction*: $p^n \to p^{n+1}$

$$\nabla \cdot \mathbf{v}^{n+1} = 0 \ ; \tag{5}$$

$$\rho\frac{\mathbf{v}^{n+1} - \mathbf{v}^*}{\Delta t} + \nabla(p^{n+1} - p^n) = 0 \ ; \tag{6}$$

which results in

$$\nabla \cdot \frac{1}{\rho}\nabla(p^{n+1} - p^n) = \frac{\nabla \cdot \mathbf{v}^*}{\Delta t} \ ; \tag{7}$$

- *Velocity correction*: $\mathbf{v}^* \to \mathbf{v}^{n+1}$

$$\mathbf{v}^{n+1} = \mathbf{v}^* - \frac{\Delta t}{\rho}\nabla(p^{n+1} - p^n) \ . \tag{8}$$

At steady state, $\mathbf{v}^* = \mathbf{v}^n = \mathbf{v}^{n+1}$, and the residuals of the pressure correction vanish, implying that the result does not depend on the time step Δt. θ denotes the implicitness factor for the viscous terms ($\theta = 1$: first order, fully implicit; $\theta = 0.5$: second order, Crank–Nicholson). One can replace the one-step explicit advective-diffusive predictor by a multistage Runge–Kutta scheme [31], allowing for higher accuracy in the advection-dominated regions and larger time steps without a noticeable increment in CPU cost. Projection schemes of this kind (explicit advection with a variety of schemes, implicit diffusion, pressure-Poisson equation for either the pressure or pressure increments) have been widely used in conjunction with spatial discretizations based on finite differences [41–44], finite volumes [45], and finite elements [2, 17, 18, 26, 27, 31, 46–51].

One complete time step is then comprised of the following substeps:

- predict velocity (advective-diffusive predictor, Equation (4);
- extrapolate the pressure (imposition of boundary conditions);
- update the pressure (Equation (7));
- correct the velocity field (Equation (8));
- if running as volume of fluid (VOF) or level set (LS) (Section 4.1.3):
  - extrapolate the velocity field and
  - update the scalar interface indicator.

4.1.2. Spatial discretization. As stated before, we desire a spatial discretization with unstructured grids in order to

- approximate arbitrary domains and
- perform adaptive refinement [73] in a straightforward manner, that is, without changes to the solver.

Given that for tetrahedral grids solvers based on edge data structures incur a much lower indirect addressing and CPU overhead than those based on element data structures [12], only these were considered. The advection operator is stabilized using edge-based upwinding [17]. The velocities required for the upwind operator are extrapolated using gradients computed at the nodes and then limited using the van Albada limiter [12].

4.1.3. Volume of fluid extensions. The extension of a solver for the incompressible Navier–Stokes equations to handle free surface flows via the VOF or LS techniques requires a series of extensions, which are the subject of the present section.

For the VOF or LS formulations, the liquid–air interface is described by a scalar equation of the form:

$$\Phi_t + \mathbf{v}_a \cdot \nabla\Phi = 0 \ . \tag{9}$$

For the classic VOF technique, Φ represents the total density of the material in a cell/element or control volume [52–56]. For pseudo-concentration techniques, Φ represents the percentage of liquid in a cell/element or control volume. For the LS approach, Φ represents the signed distance to the interface [57, 58]. Combinations of these approaches were also proposed [59]. The same scheme is used for the transport equation that describes the propagation of the VOF fraction, pseudo-concentration, or distance to the free surface given by Equation (9).

Before going on, we remark that both the VOF and LS approaches are implemented in the code. Experience indicates that both work well. For VOF, the profiles can be sharp fronts that propagate through the domain. This places a higher demand on the quality of the advection schemes used to integrate Equation (9). The divergence operator is stabilized using a fourth-order pressure damping [12]. In particular, it is important to have a monotonicity preserving scheme for Φ. The profiles advected for LS are smoother and do not place such demands on advection schemes. On the other hand, for LS, it is important to balance the cost and accuracy loss of reinitializations vis-à-vis propagation. Given that the advection solvers used are all monotonicity preserving, and that the VOF option is less CPU demanding than LS, only the VOF technique is considered in the following.

The pressure and velocities in the air region are extrapolated in a consistent way in order to obtain proper values in the air region close to the free surface [20].

Experience indicates that the amount of liquid mass (as measured by the region where the VOF indicator is larger than a cut-off value) does not remain constant for typical runs. The reasons for this loss or gain of mass are manifold: loss of steepness in the interface region, inexact divergence of the velocity field, boundary velocities, and so on. This lack of exact conservation of liquid mass has been reported repeatedly in the literature [57–62]. The recourse taken here is the classic one: add/remove mass in the interface region in order to obtain an exact conservation of mass [20].

Given that the air region is not treated/updated, any CPU spent on it may be considered as wasted. Most of the flow solver work is spent in loops over the edges (upwind solvers, limiters, gradients, etc.). As the edges have to be grouped in order to avoid memory contention/allow vectorization when forming right-hand sides [63, 64], this opens a natural way of avoiding unnecessary work: form relatively small edge groups that still allow for efficient vectorization and deactivate groups instead of individual edges [12]. In this way, the basic loops over edges do not require any changes. The `if` test whether an edge group is active or deactive is placed outside the inner loops over edges, leaving them unaffected. On scalar processors, edges groups as small as eight are used. Furthermore, if points and edges are grouped together in such a way that proximity in memory mirrors spatial proximity, most of the edges in air will not incur any CPU penalty.

4.1.4. Treatment of moving bodies. Moving bodies may be treated in a variety of ways.

The first one considers a body-fitted mesh that is moved according to the body motion. This requires a consistent update of geometrical quantities (mass, shape-functions, etc.) in order to satisfy the discrete geometric conservation law [65] algorithms to move the mesh that minimizes the number of distorted elements [3] and automatic remeshing during the run [63]. We consider this approach to be the most accurate and have always used it as the starting point to compute the cases shown in the succeeding paragraphs. This approach is used here.

The second approach considers a mesh that covers the whole domain. The (surfaces of the) bodies are embedded in this mesh and move through it [12, 18, 33]. This requires the recalculation of all edges cut by the embedded surfaces [18], a consistent update of boundary conditions, geometrical quantities (mass, shape-functions, etc.), and the deactive region inside the bodies, as well as the proper calculation of forces on the embedded surfaces.

The third approach also considers a mesh that covers the whole domain. However, in this case, bodies are immersed in this mesh and move through it [12]. When one applies proper kinematic or kinetic boundary conditions at the points covered by the immersed body, the fluid is updated as if a hard surface was present. Note that unlike the previous two approaches, in this case, the flowfield is also updated inside the moving bodies. This approach requires the recalculation of all points covered by the immersed bodies and a consistent update of boundary conditions and forces transmitted to the bodies.

4.2. Mixed interface-tracking/interface-capturing technique

The main features of this numerical approach are as follows:

- Monolithic fully coupled solution for the Navier–Stokes problem, system of Equations (1) and (2) [66, 67].
- Mixing interface capturing and tracking techniques to describe interfaces: the free surface and the rigid-body contour (see [22] and references therein).
- Fixed-mesh technique tested in structured and unstructured grids.
- Local mass conservation algorithms were developed.
- Sub-elemental integrations are applied to capture property discontinuities.
- Fully implicit time integration.

4.2.1. Basic elements of the solver. The Navier–Stokes equations are solved over the entire spatial domain where we have one solid and two fluids. The interfaces between the different regions are determined during the computation. The motion of those interfaces and assignment (i.e., distribution) of the material properties in different regions of the domain are coupled with the solution of Equations (1) and (2). The Navier–Stokes equations are discretized with the FEM and the generalized streamline operator technique [66, 67]. The stabilized nature of the technique allows the use of equal-order interpolation functions for velocity and pressure. In the context of flow analysis with moving interfaces, all the element-level matrices and vectors are computed by taking into account the discontinuities in the material properties. The time integration is performed using a standard backward Euler scheme.

DOI: 10.1002/fld

4.2.2. Interface update. In the numerical model based on the MITICT [19,23], we handle the fluid–solid interfaces with the MLIT [68,69] and the fluid–fluid interfaces with the ETILT [21,23,34]. The MLIT was shown to be robust in preserving the interface geometry, with the numerical results also showing independence from the element and time-step sizes [68, 69]. These features are important in the accurate representation of the fluid–solid interface because, despite the large displacements of the interface points, we need to preserve the interface geometry. The MLIT performs very well in that respect. The ETILT, on the other hand, is capable of representing the fluid–fluid interfaces in a robust fashion, even when those interfaces undergo large distortions. We mix the MLIT and ETILT as follows:

(a) The MLIT has priority in the motion of the fluid–solid interface and the distribution of the material properties. We assume a rigid-body motion for the solid. The volume occupied by the rigid body and its shape do not change. The material properties are distributed in the solid and fluid (including fluids 1 and 2) regions according to the boundaries defined by the fluid–solid and fluid–fluid interfaces.

(b) The ETILT needs to recognize the fluid–solid interface described by the MLIT In the fluid, the property distribution is governed by the ETILT, but the body is immersed in either fluid 1 or 2 or both. As the fluid–fluid interface is updated after moving the fluid–solid interface, it is necessary to avoid overlap between the solid and new fluid distribution and, in addition, to preserve the volume in chunks of fluids 1 and 2.

The main aspects of these techniques are briefly described below, together with some details on how the MLIT and ETILT are coupled.

Fluid–solid interface

The fluid–solid interface is defined by a collection of points that serve as markers. The motion of that material front is described by updating the markers positions according to a simple Lagrangian scheme:

$$\mathbf{x}^{t+\Delta t} = \mathbf{x}^{t} + \mathbf{v}^{t+\theta\Delta t}\Delta t \quad . \tag{10}$$

where Δt is the time step, θ is the time integration parameter (usually set to 1.0), $\mathbf{x}^t$ is the known vector of markers positions at time t, $\mathbf{x}^{t+\Delta t}$ is the updated position at time $t+\Delta t$, and $\mathbf{v}^{t+\theta\Delta t}$ is the velocity of the markers. To avoid loss of mass during the interface evolution, we combined a global mass correction algorithm with the solution of Equation (10). Material properties are assigned at integration-point level, depending on what side of the interface the integration point belongs to. To capture the discontinuity in material properties, we subdivided the elements crossed by the interface to have higher-resolution integration. The integration strategy adopted in this work consists in subdividing the element parent domain by taking uniform length subintervals in each space direction (in practice, three partitions in each direction are used). No regularization is needed in this field. Details and numerical evaluations of the MLIT can be found elsewhere [68, 69].

Fluid–fluid interface

To identify the interface between the two fluids, we used a Heaviside function H. In the discrete space, an indicator along the edges of the elements crossed by the interface can be used to define the interface position. In the ETILT, $\Phi^{h_e} = H(\Phi^h(\mathbf{x}_f))$, where Φ^{h_e} is the edge-based representation of the interface, $\mathbf{x}_f$ is the interface position, and Φ^h is the nodal representation of Φ. Any value of z within the interval [0,1] could be considered provided that all the steps involved in the ETILT be consistently performed with such a value [21, 34]. Here, the choice $z = 0.5$ is adopted to mark the interface. In this context, Φ is computed by using a constrained least-squares projection [19,21,72]:

$$\int \Psi^h(\Phi_t^h - \Phi_t^{h_e}) + \sum_{k=1}^{n_{\mathrm{ie}}} \lambda_{\mathrm{pen}}\Psi^h(\mathbf{x}_x)(\Phi_t^h(\mathbf{x}_k) - 0.5) = 0 \tag{11}$$

where Ψ^h is the test function, n_{ie} is the number of element edges crossed by the interface, $\mathbf{x}_x$ is the coordinate of the interface location along the kth edge, and λ_{pen} is a penalty parameter. After this projection, the Φ^h at time $t + \Delta t$, that is, $\Phi^h_{t+\Delta t}$, is computed using a discrete formulation of the advection equation (Equation (9)) on the basis of the streamline upwind/Petrov–Galerkin formulation [70, 71]. $\Phi^{h_e}_{t+\Delta t}$ is obtained from $\Phi^h_{t+\Delta t}$ by using a combination of a least-squares projection and corrections to enforce volume conservation for chunks of fluids 1 and 2. At each time step, a property P is interpolated as follows:

$$P^h = \Phi^{h_e} P_1 + (1 - \Phi^{h_e}_t) P_2 \ . \tag{12}$$

As in the MLIT, the elements crossed by the interface are integrated with higher-resolution integration to capture the discontinuity in material properties, and no regularization is needed in the P field. Details and numerical evaluation of the ETILT can be found elsewhere [21, 34]. We note that for the present study, $\Phi^{h_e}_{t+\Delta t}$ must additionally identify the rigid zone, that is, the fluid–solid interface. This means that the condition $(\Phi^h_{t+\Delta t} - 0.5)$ needs to be strongly satisfied for points belonging to that interface. To this end, the edges crossed by the Lagrangian fluid–solid interface are added to the summation in Equation (11). Moreover, as the rigid-body motion has priority and the computation via the MLIT ensures the volume preservation in the solid, the ETILT corrections in $\Phi^{h_e}_{t+\Delta t}$ to enforce volume conservation are not used along the fluid–solid interface. This strategy handles the overlap in the zone of action of the two methods when the rigid body is simultaneously in touch with fluids 1 and 2.

5. ESTIMATION OF NUMERICAL ERRORS

In order to estimate the element size distribution in space required to obtain a sufficiently accurate result, we carried out a whole series of mesh convergence studies. The cases considered followed the two regimes expected to be critical for the flow: settling velocity regime and splashdown regime. As the first of the flow clode employed is 3D, a 30° section of the tube was computed. The geometry, together with the boundary conditions, may be seen from Figure 6.

For the settling regime, an incoming velocity of $v = 0.001$ m/s was considered, whereas for the splashdown regime, the velocity was set to $v = 1.000$ m/s. This yields Reynolds numbers of the order of $Re_1 = 0.1952$ and $Re_2 = 195.2$, respectively. As the motion of the sphere is determined by the resulting forces, the vertical force F_y was considered as the convergence criterion. In order to ascertain the quality of the pure dynamic pressure and viscous forces, we imposed no gravity field on the flow. Both isotropic and non-isotropic (so-called RANS) grids were considered. A zoom of the region near the sphere in the plane $z = 0$ showing the different types of grids is plotted in Figure 7.

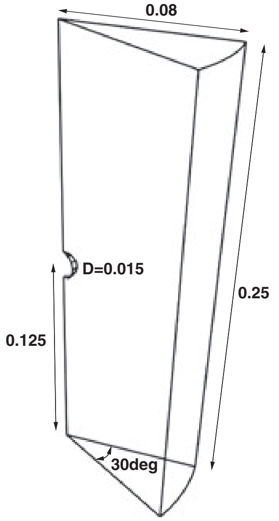


Figure 6. Geometry and boundary conditions for error estimation (dimensions in m).

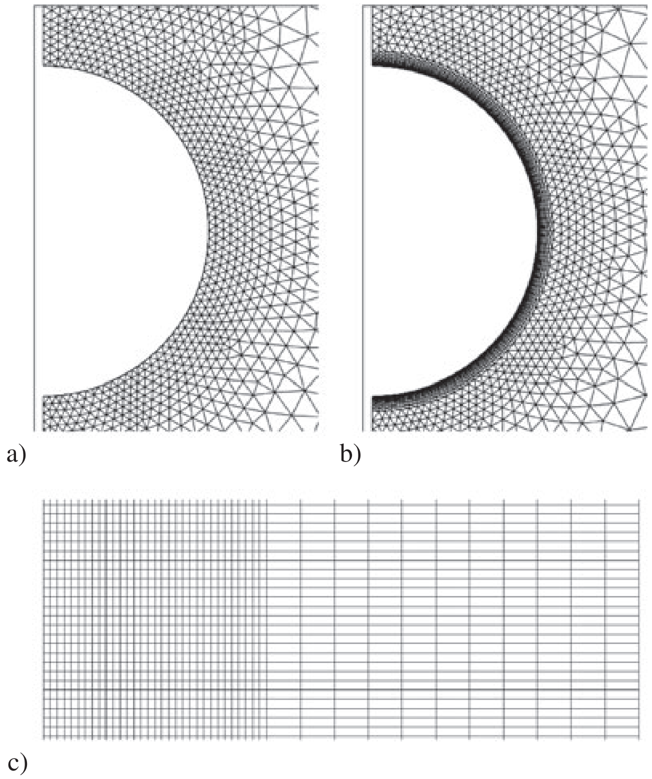


Figure 7. Meshes adopted in the present work, for body-fitted technique: (a) isotropic and (b) non-isotropic mesh in the vicinity of the sphere; for MITICT (c) zoom of the non-uniform regular fixed quadrilateral mesh.

The results for the isotropic and non-isotropic meshed with $h_{\mathrm{iso}} = 0.375 \cdot 10^{-3}$ and $Re_2 = 195.2$ are shown in Figure 8. The countour lines are the same, and 60 contours were used. Any discrepancy between the results is therefore of the order of 1–2%. Tables I and II summarize the results obtained. Here, h_{iso} denotes the (uniformly small) isotropic element size near the sphere; $h_{\min}$, the smallest edge length in the mesh; nlayr, the number of semi-structured layers of elements near the boundary; and nelem, the total number of elements in the domain.

The equivalent drag coefficients obtained are $c_{\mathrm{d}1} = 154.0$ and $c_{\mathrm{d}2} = 0.8326$. For a free-standing sphere, the experimental values may be approximated by the following formula:

$$c_{\mathrm{d}} = \frac{24}{Re}\left(1 + 0.15\, Re^{0.687}\right). \tag{13}$$

This yields $c_{\mathrm{d1exp}} = 128.9$ and $c_{\mathrm{d2exp}} = 0.8138$, respectively. These do not differ significantly from the results obtained here, but one has to bear in mind that the boundaries of the tube, where no-slip condition is also imposed, are rather close. Convergence studies conducted by the authors showed that even quadrupling the tube diameter and doubling the tube length still had an effect on the forces obtained.

One can see that for an accuracy comparable with that of the experimental observations, an isotropic mesh with proper wake resolution and a minimum element size of $h = 0.375 \cdot 10^{-3}$ m close to the sphere should be sufficient. The next question that arose was how to quantify the numerical error in the buoyancy forces. After all, the final settling velocity will depend crucially on whether buoyancy and gravity/mass balance out properly. In order to ascertain this error, we assumed that the sphere had the same density as the liquid. The total resulting force acting on the sphere in a field with gravity should then be the same as the force obtained without gravity. Here, after a frustrating series of tests, it was seen that it is crucially important to set the mass of the sphere to be that obtained from the displaced fluid volume. Because of the finite resolution of the sphere surface with

DOI: 10.1002/fld

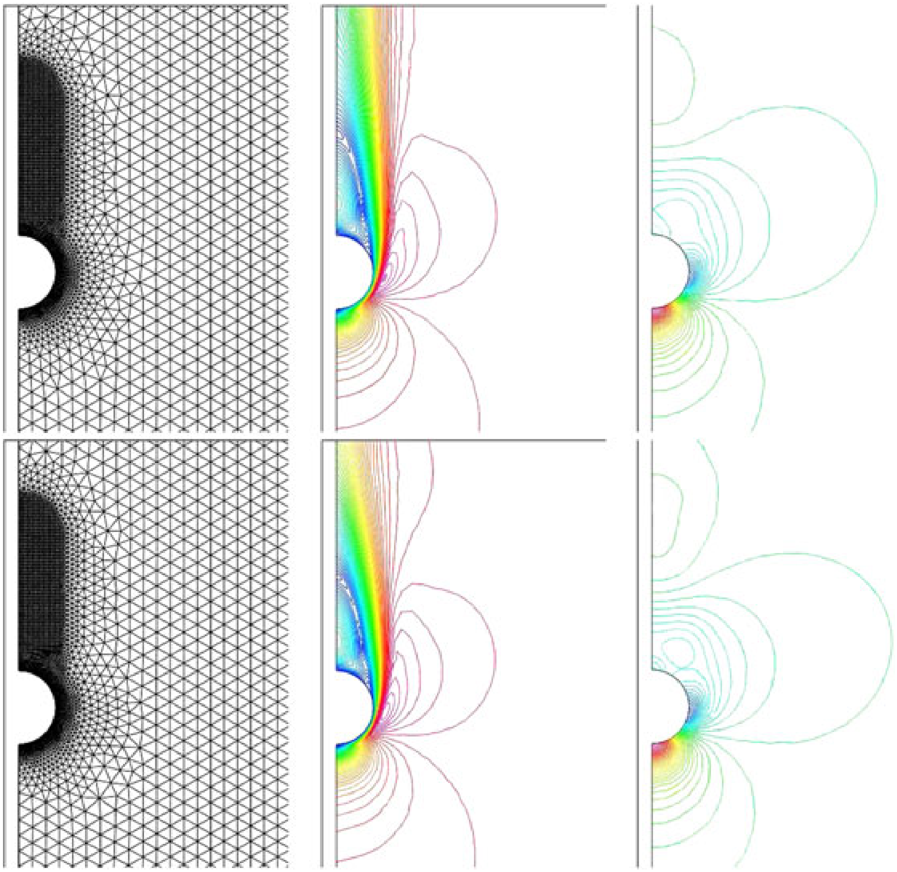


Figure 8. Results for isotropic and non-isotropic grids.

Table I. $Re = 0.1952$ (settling regime).

$10^3\ h_{\text{iso}}$	$10^3\ h_{\text{min}}$	`nlayr`	`nelem`	F_y	Error (%)
0.750	0.7500	0	27,272	1.003	−2.90
0.375	0.3750	0	205,077	1.008	−2.42
0.187	0.1870	0	1,622,456	1.013	−1.94
0.375	0.0410	9	220,957	1.030	−0.29
0.750	0.0085	18	38,702	1.036	0.29
0.375	0.0066	18	245,507	1.033	0.00

Table II. $Re = 195.2$ (splashdown regime).

$10^3\ h_{\text{iso}}$	$10^3\ h_{\text{min}}$	`nlayr`	`nelem`	$10^3\ F_y$	Error (%)
0.750	0.7500	0	27,272	5.161	−7.59
0.375	0.3750	0	205,077	5.212	−6.68
0.187	0.1870	0	1,622,456	5.353	−4.15
0.375	0.0410	9	220,957	5.451	−2.40
0.750	0.0085	18	38,702	5.576	−0.16
0.375	0.0066	18	245,507	5.585	0.00

triangles, this is not the same as the exact volume of the sphere. For the grids used here, the errors have been compiled in Table III. The reduction in error in not uniform, as expected, because of the surface representation used. Note that although the error in volume is small, in the settling regime, the other forces are small too, so any error here can lead to very large errors in the final velocity.

DOI: 10.1002/fld

Table III. Error in volume.

$10^3\ h_{\mathrm{iso}}$	$10^7\ \Omega$	Error (%)
0.750	1.464696	−0.54
0.375	1.469463	−0.21
0.187	1.470650	−0.13
0.000	1.472621	0.00

6. COMPARISON RUNS

After making sure that the numerical errors would be in the range of the experiments, a series of runs was performed. As it was mentioned in Section 2, at least for spheres falling into oil, the video images obtained showed no sign of 3D effects (e.g., sphere rotation), so this possibility was discarded in the numerical simulations, and, consequently, an axisymmetric model is used in the numerical analysis. To build such a model, we obtained the body-fitted computations in a 3D spatial model (as it is shown in Figure 6) by using tetrahedral elements but mimic axisymmetric conditions by accordingly imposing the boundary conditions. Hence, 3D effects cannot be reproduced in such a model. The MITICT calculations are performed in a plane 2D axisymmetric model where four noded isoparametric elements are used. The mesh size and time step used in the MITICT analysis are fixed as $h = 0.635$ mm and $\Delta t = 5 \cdot 10^{-4}$ s considering these values as characteristics values obtained from the analysis reported in the previous section.

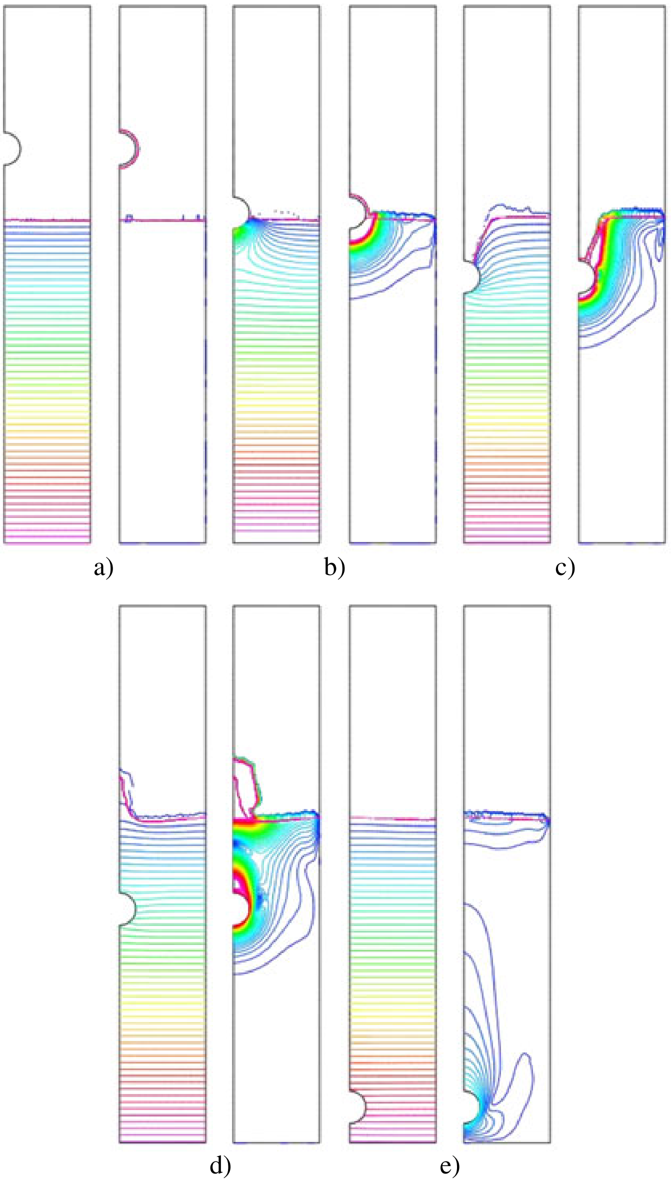
6.1. Silicone sphere falling into oil

The parameters for the first case were as follows:

- diameter of sphere: $d = 0.015$ m;
- density of sphere: $\rho_s = 962$ kg/m^3;
- diameter of tube: $d = 0.080$ m;
- length of tube: $H = 0.25$ m;
- density of liquid (oil): $\rho_s = 911$ kg/m^3;
- viscosity of liquid (oil): $\mu_l = 0.07$ kg/m/s;
- fall distance for sphere: $L = 0.08$ m.

A few images obtained using body fitted have been compiled in Figure 9 where the pressure and velocity modulus contours are plotted at times $t = 0.1, 0.15, 0.2, 0.25$, and 1.0 s. Figure 10 shows the same pressure and velocity modulus contours computed by MITICT at the same instants (note that the contours in the air phase are only present in Figure 10 as the body-fitted solution does not solve the air phase whereas MITICT does). An overall good agreement between both formulations can be observed. The comparison of the computed position history with the experimental measurements is shown in Figure 11. The numerical predictions using different approaches and mesh sizes are in good correlation between them and with the experiments. Particularly, the different regions of the motion are clearly captured: gravity-driven motion, impingement with the free surface and viscous-driven motion in oil.

Note that at first glance, the comparison of velocities in Figure 12 is very satisfactory. However, a zoom into the time interval $0.5 < T < 1.75$ reveals that there is a systematic discrepancy of approximately 10% with the experiment in this 'creeping flow' region. The terminal settling velocity for a free-standing sphere under the same conditions is $v_s = 0.0515$ m/s, which corresponds to a Reynolds number of $Re_s = 10.1$. The numerical results are clearly in this range. Figure 13 shows the experimental data of the air–oil free surface evolution (average value taken from the experiments) from the sphere impingement up to the rest of the surface together with the free surface height computed using both techniques. The numerical solutions satisfactorily matches the time evolution of the wave. The amplitude is particularly well captured by the body-fitted formulation, which could be expected because of the mesh refinement in the free surface region to solve Equation (9).



a) b) c)

d) e)

Figure 9. Silicone sphere falling into oil, body-fitted solutions: pressure (50 contours from 0 to 1340 Pa) and velocity modulus (50 contours from 0 to 0.2 m/s) at times (a) 0.10, (b) 0.15, (c) 0.20, (d) 0.25, and (e) 1.00 s.

6.2. *Glass sphere falling into oil*

The parameters for the second case were as follows:

- diameter of sphere: $d = 0.015$ m;
- density of sphere: $\rho_s = 3594$ kg/m^3;
- diameter of tube: $d = 0.080$ m;
- length of tube: $H = 0.25$ m;
- density of liquid (oil): $\rho_s = 911$ kg/m^3;
- viscosity of liquid (oil): $\mu_l = 0.07$ kg/m/s;
- fall distance for sphere: $L = 0.08$ m.

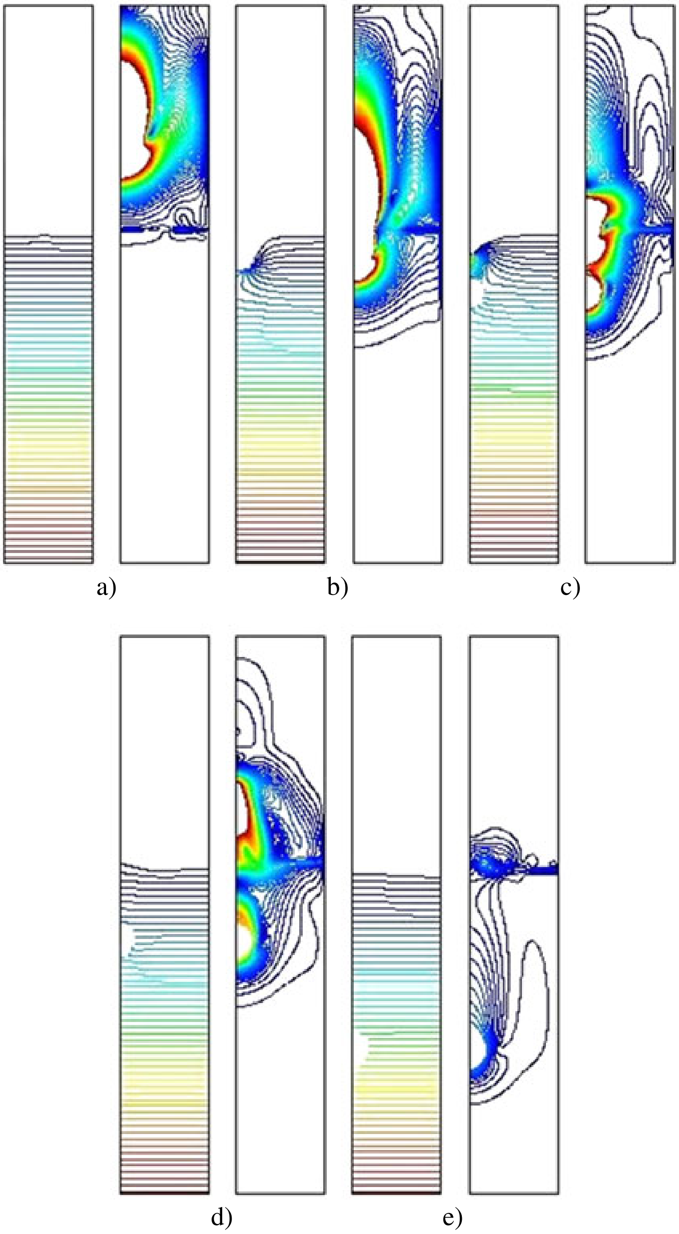


Figure 10. Silicone sphere falling into oil, MITICT solutions: pressure (50 contours from 0 to 1340 Pa) and velocity modulus (50 contours from 0 to 0.2 m/s) at times (a) 0.10, (b) 0.15, (c) 0.20, (d) 0.25, and (e) 1.00 s.

A few images obtained using body fitted have been compiled in Figure 14 where pressure and velocity modulus contours are plotted at times $t = 0.1, 0.15, 0.2, 0.25$, and 0.35 s. Figure 15 shows the same pressure and velocity modulus contours computed by MITICT at the same instants.

Pressure and velocity modulus countours present similar trends to those reported in Figures 9 and 10. The comparison of the sphere motion with experiments is shown in Figure 16. In spite of the experimental error in time, a good qualitative description of the position history is obtained. The numerical responses are in good agreement between them. The predicted body velocity is presented in Figure 17. The terminal settling velocity for a free-standing sphere under the same conditions is $v_s = 0.8047$ m/s, which corresponds to a Reynolds number of $Re_s = 157$. As before, the numerical results are clearly in this range. The oscillations are due to the collapse of the air bubble above the

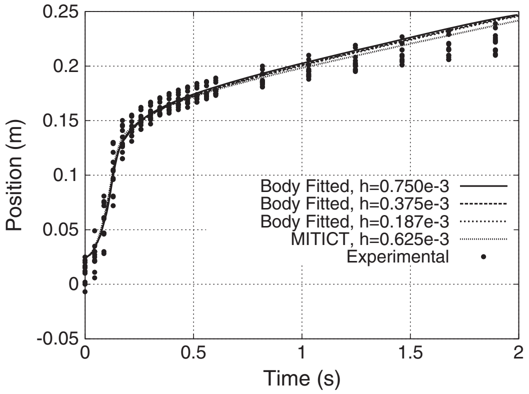


Figure 11. Silicone sphere falling into oil: comparison of position history.

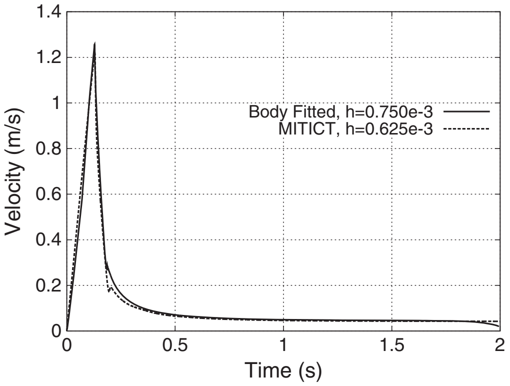


Figure 12. Silicone sphere falling into oil: comparison of velocity history.

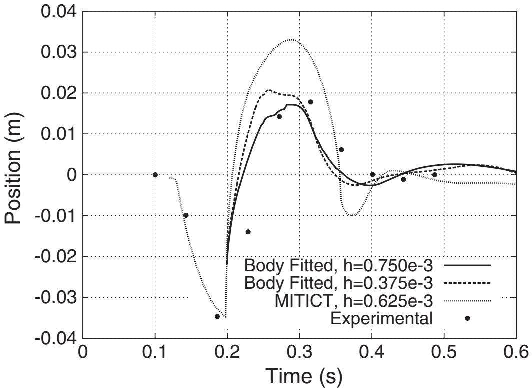


Figure 13. Silicone sphere falling into oil: comparison of free surface history.

sphere. This collapse produces considerable pressures, which are reflected in the velocities. Once this overpressure dissipates, the velocity returns to the expected smooth, viscous behavior. The free surface motion cannot be described quantitatively from the experiments because of the error in time. Nevertheless, its computed evolution is plotted in Figure 18.

DOI: 10.1002/fld

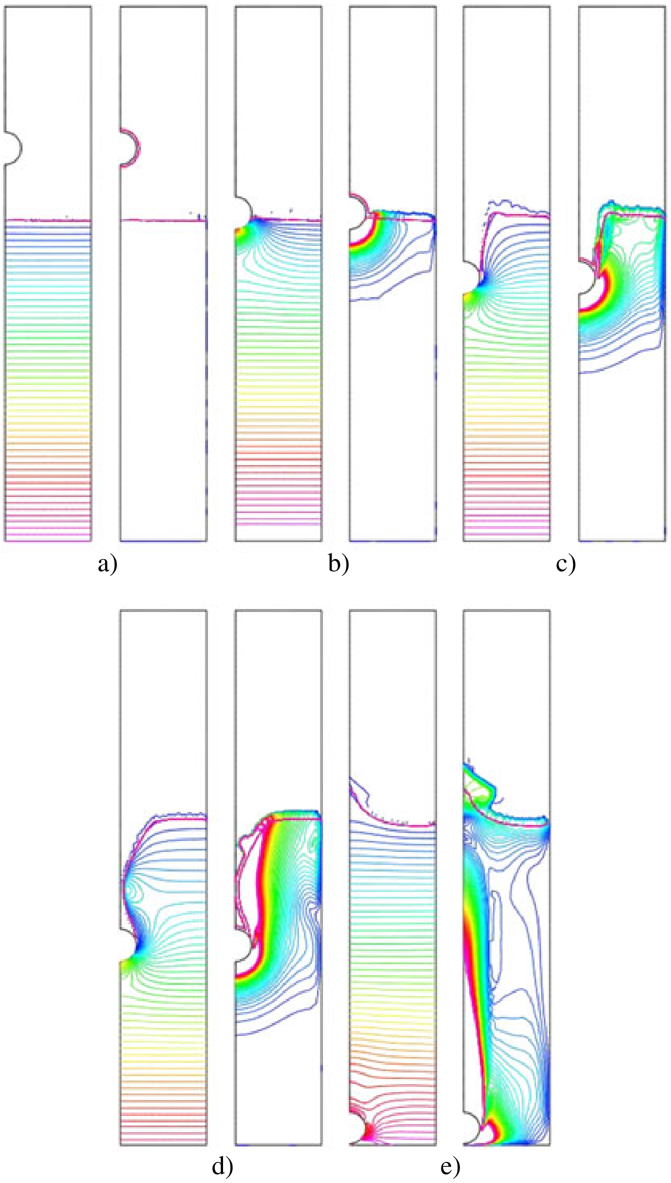


Figure 14. Glass sphere falling into oil, body-fitted solutions: pressure (50 contours from 0 to 1340 Pa) and velocity modulus (50 contours from 0 to 0.2 m/s) at times (a) 0.10, (b) 0.15, (c) 0.20, (d) 0.25, and (e) 0.35 s.

6.3. Silicone sphere falling into water

The parameters for this third case were as follows:

- diameter of sphere: $d = 0.015$ m;
- density of sphere: $\rho_s = 962$ kg/m^3;
- diameter of tube: $d = 0.080$ m;
- length of tube: $H = 0.25$ m;
- density of liquid (water): $\rho_s = 1000$ kg/m^3;
- viscosity of liquid (water): $\mu_l = 0.07$ kg/m/s;
- fall distance for sphere: $L = 0.08$ m.

A few images obtained using body fitted have been compiled in Figure 19 where the pressure and velocity modulus contours are plotted at times $t = 0.10, 0.15, 0.20, 0.25, 0.50, 0.75, 1.00, 1.50,$ and

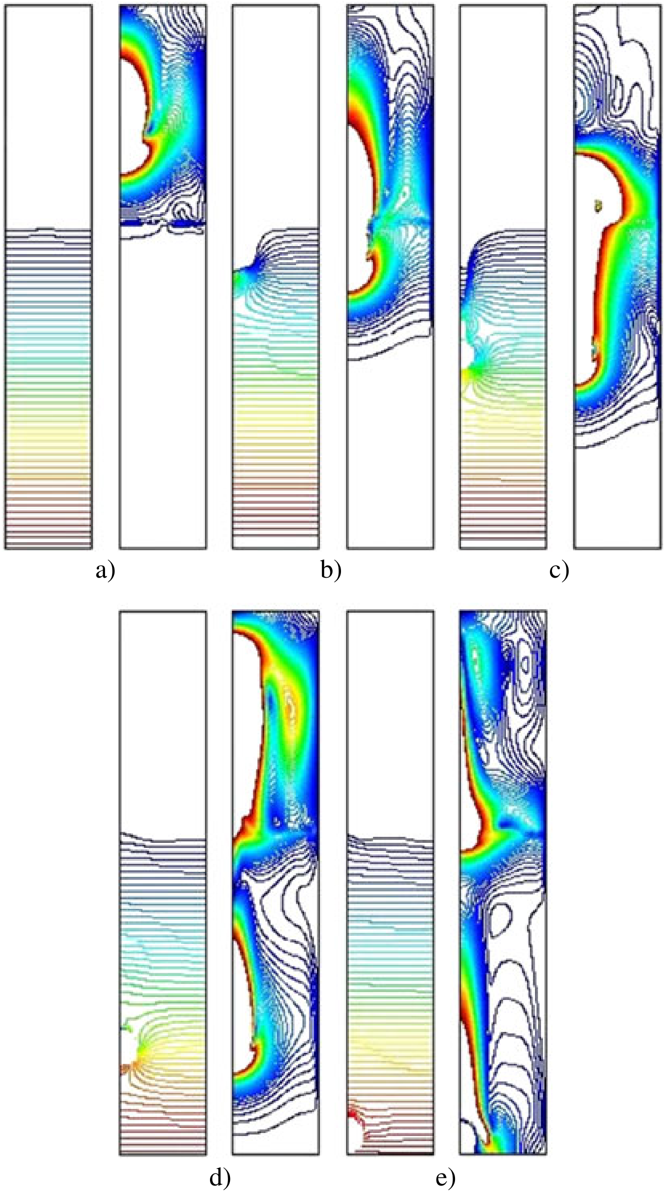


Figure 15. Glass sphere falling into oil, MITICT solutions: pressure (50 contours from 0 to 1340 Pa) and velocity modulus (50 contours from 0 to 0.2 m/s) at times (a) 0.10, (b) 0.15, (c) 0.20, (d) 0.25, and (e) 0.35 s.

2.00 s. Figure 20 shows the same pressure and velocity modulus contours computed by MITICT at the same instants.

As mentioned already, the experimental error in time do not allow a quantitative comparison with them. The numerical responses are presented in Figures 21 and 22 for position and velocity histories of the sphere, respectively. In this case, major quantitative differences are obtained for the numerical responses. The numerical discrepancies can be attributed to the more complex flow pattern present in this case, that is, the sphere moves initially downwards but around $t = 1$ s, it starts going up. We believe that viscosity effects around the sphere walls may play a relevant role in this case. It is seen that in this case, this localized phenomenon is not properly captured by the simulations where, in particular, the MITICT solution exhibits the largest numerical–experimental difference denoting also high diffusive effects (due to the high viscosity adopted in the numerical computation of this case). The analysis of this boundary layer is a subject of future research.

DOI: 10.1002/fld

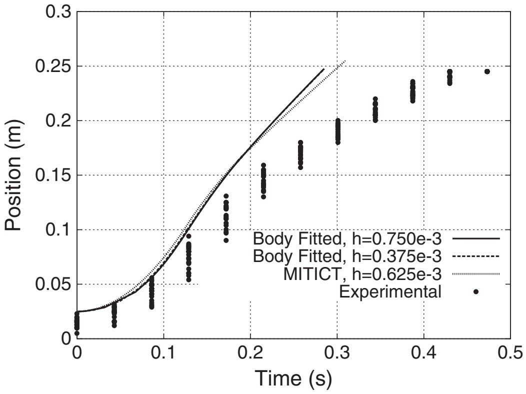


Figure 16. Glass sphere falling into oil: comparison of position history.

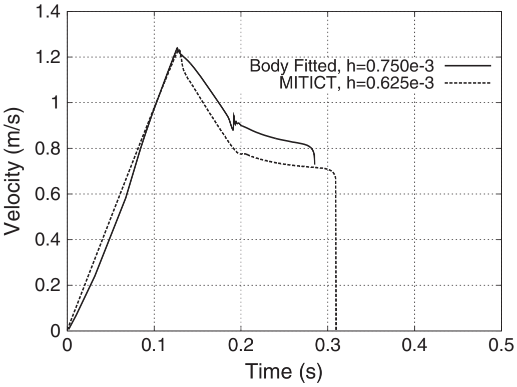


Figure 17. Glass sphere falling into oil: comparison of velocity history.

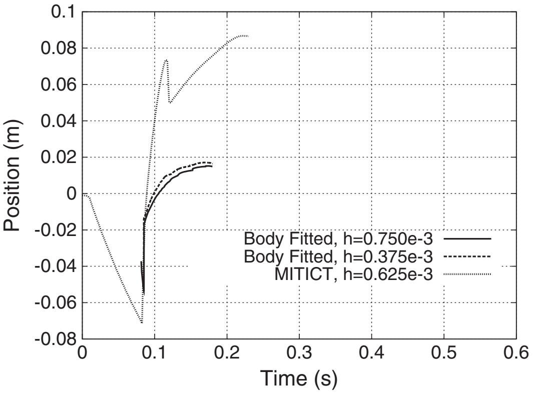


Figure 18. Glass sphere falling into oil: comparison of free surface history.

7. CONCLUSIONS AND OUTLOOK

The experimental and numerical analysis of spheres falling into viscous flows was considered. The physical model was built using a set of spheres of silicone and glass falling into oil and water. The rigid-body trajectory of the spheres and the free surface evolution were obtained from videos. The experimental error was estimated, and a whole series of experimental runs was conducted to obtain reliable measurements. Two different codes were used to obtain the numerical results. The first uses a fractional step approach with adaptive meshes and time-step sizes, whereas the second

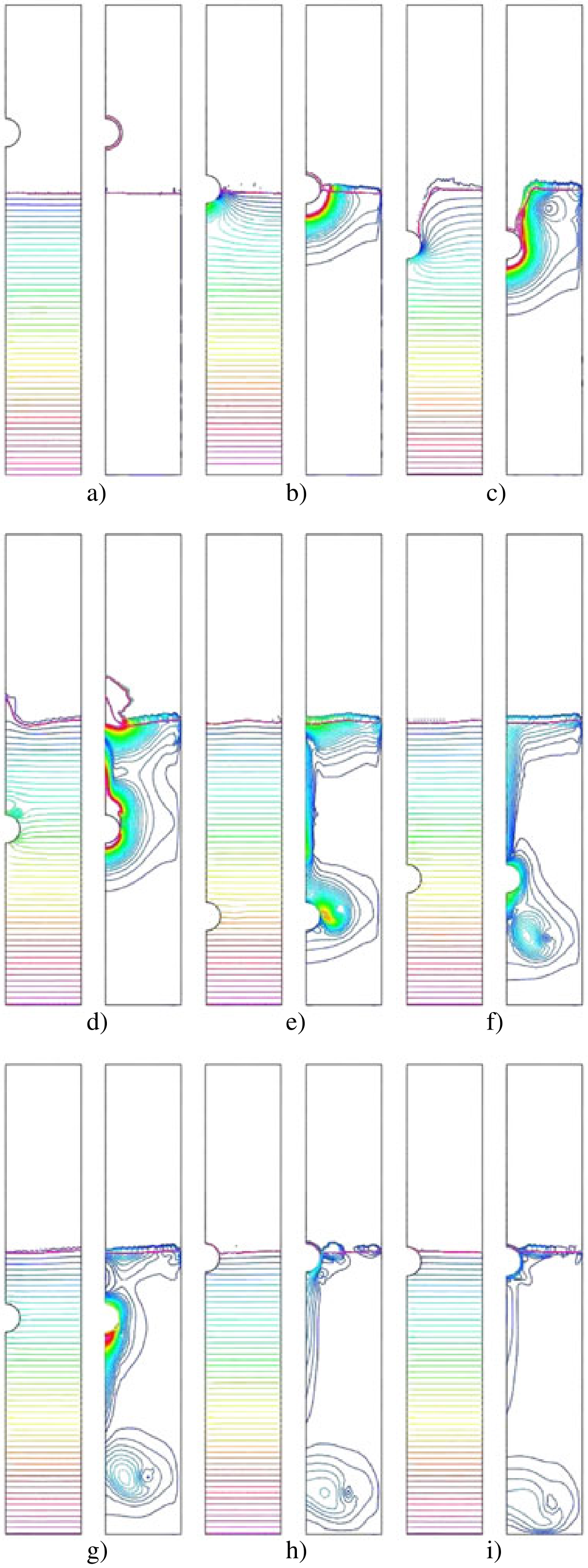


Figure 19. Silicone sphere falling into water, body-fitted solutions: pressure (50 contours from 0 to 1340 Pa) and velocity modulus (50 contours from 0 to 0.2 m/s) at times (a) 0.10, (b) 0.15, (c) 0.20, (d) 0.25, (e) 0.50, (f) 0.75, (g) 1.00, (h) 1.50, and (i) 2.00 s.

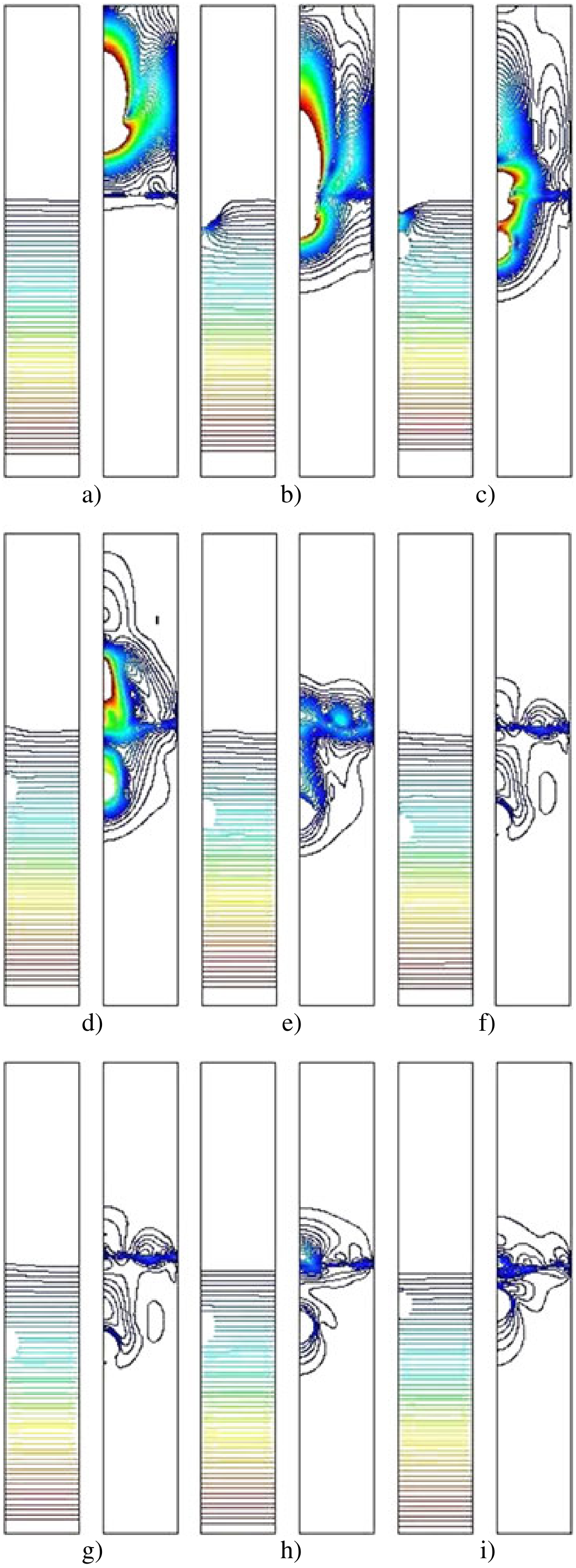


Figure 20. Silicone sphere falling into water, MITICT solutions: pressure (50 contours from 0 to 1340 Pa) and velocity modulus (50 contours from 0 to 0.2 m/s) at times (a) 0.10, (b) 0.15, (c) 0.20, (d) 0.25, (e) 0.50, (f) 0.75, (g) 1.00, (h) 1.50, and (i) 2.00 s.

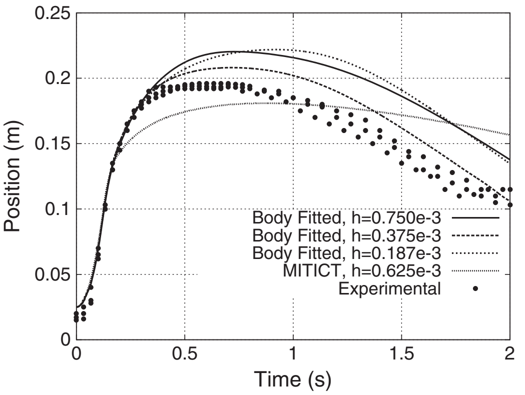


Figure 21. Silicone sphere falling into water: comparison of position history.

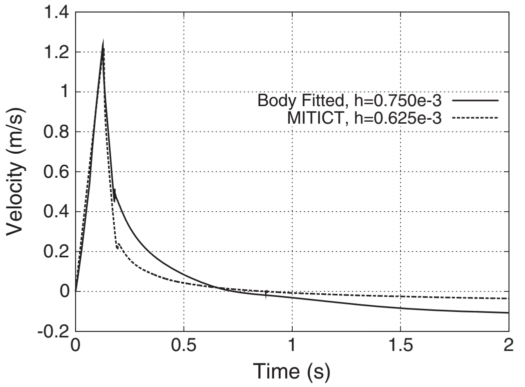


Figure 22. Silicone sphere falling into water: comparison of velocity history.

uses a monolithic fully coupled fixed-mesh technique. A convergence study was carried out to ascertain the numerical errors. The grids used were such that the error was of the same order of magnitude as the error estimated for the experiments. Three cases were considered in detail: a silicone sphere falling into oil, a glass sphere falling into oil, and a silicone sphere falling into water. The results obtained exhibit a good comparison between both numerical techniques and with the experiments particularly those related to low Reynolds number, that is, silicone–oil case. As expected, the body-fitted technique, which matches the body shape at any time of the analyses, reasonably adjusts the body motion for all cases. In spite of the poor representation of the boundary layer around the rigid body, the MITICT results show a satisfactory numerical response. Mesh sizes refinement near the rigid body when using such technique as well as the bubble effects and 3D behaviors need to be explored in future works.

ACKNOWLEDGEMENTS

The supports provided by DICYT (USACH) and CONICYT Fondecyt 1095028 are gratefully acknowledged. A considerable amount of the reported work was carried during the stay of the second author at the University of Santiago de Chile in January 2010.

REFERENCES

1. Hughes TJR, Liu WK, Zimmermann TK. Lagrangian–Eulerian finite element formulation for incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering* 1981; **29**:329–349.
2. Löhner R. A Fast Finite Element Solver for Incompressible Flows. *AIAA-99-0398* 1990.

DOI: 10.1002/fld

3. Löhner R, Yang C. Improved ALE mesh velocities for moving bodies. *Communications in Numerical Methods in Engineering* 1996; **12**:599–608.
4. Sarrate J, Huerta A, Donea J. Arbitrary Lagrangian–Eulerian Formulation for fluid-rigid body interaction. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:3171–3188.
5. Dettmer W, Perić D. A computational framework for fluid–structure interaction: finite element formulation and applications. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:5754–5779.
6. Khurram RA, Masud A. A multiscale/stabilized formulation of the incompressible Navier–Stokes equations for moving boundary flows and fluid–structure interaction. *Computational Mechanics* 2006; **38**:403–416.
7. Masud A. Effects of mesh motion on the stability and convergence of ALE based formulations for moving boundary flows. *Computational Mechanics* 2006; **38**:430–439.
8. Tezduyar TE, Behr M, Liou J. 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. *Computer Methods in Applied Mechanics and Engineering* 1992; **94**:339–351.
9. Tezduyar TE, Behr M, Mittal S, Liou J. A new strategy for finite element computations involving moving boundaries and interfaces—the deforming-spatial-domain/space–time procedure: II. Computation of free-surface flows, two-liquid flows, and flows with drifting cylinder. *Computer Methods in Applied Mechanics and Engineering* 1992; **94**:353–371.
10. Tezduyar T, Sathe S, Stein K. Solution techniques for the fully discretized equations in computation of fluid–structure interactions with the space–time formulations. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:5743–5753.
11. Idelsohn S, Marti J, Limache A, Oñate E. Unified Lagrangian formulation for elastic solids and incompressible fluids: application to fluid–structure interaction problems via the PFEM. *Computer Methods in Applied Mechanics and Engineering* 2008; **197**:1762–1776.
12. Löhner R. *Applied CFD Techniques: An Introduction Based on Finite Element Methods*, 2nd Edition. John Wiley & Sons, 2008.
13. Mori Y, Peskin Ch. Implicit second-order immersed boundary methods with boundary mass. *Computer Methods in Applied Mechanics and Engineering* 2008; **197**:2049–2067.
14. Cruchaga MA, Muñoz C, Celentano DJ. Simulation and experimental validation of the motion of immersed rigid bodies in viscous flows. *Computer Methods in Applied Mechanics and Engineering* 2008; **197**:2823–2835.
15. Yang C, Löhner R. Fully nonlinear ship wave calculation using unstructured grids and parallel computing. *Proceedings 3rd Osaka Colloquium on Advanced CFD Applications to Ship Flow and Hull Form Design*, Osaka, Japan, 1998; 125–150.
16. Yang C, Löhner R, Lu H. An unstructured-grid based volume-of-fluid method for extreme wave and freely-floating structure interactions. *Journal of Hydrodynamics* 2006; **18**:415–422.
17. Löhner R, Yang C, Oñate E, Idelssohn S. An unstructured grid-based, parallel free surface solver. *Applied Numerical Mathematics* 1999; **31**:271–293.
18. Löhner R, Baum JD, Mestreau E, Sharov D, Charman C, Pelessone D. Adaptive embedded unstructured grid methods. *International Journal for Numerical Methods in Engineering* 2004; **60**:641–660.
19. Tezduyar T. Interface-tracking and interface-capturing techniques for finite element computation of moving boundaries and interfaces. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:2983–3000.
20. Löhner R, Yang C, Oñate E. On the simulation of flows with violent free surface motion. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:5597–5620.
21. Cruchaga MA, Celentano DJ, Tezduyar TE. Collapse of a liquid column: numerical simulation and experimental validation. *Computational Mechanics* 2007; **39**:453–476.
22. Cruchaga MA, Celentano DJ, Tezduyar TE. A numerical model based on the mixed interface-tracking/interface-capturing technique (MITICT) for flows with fluid–solid and fluid–fluid interfaces. *International Journal for Numerical Methods in Fluids* 2007; **54**:1021–1030.
23. Tezduyar TE. Finite elements methods for flows problems with moving boundaries and interfaces. *Archives on Computational Methods in Engineering* 2001; **8**:83–130.
24. Akin JE, Tezduyar TE, Ungor M. Computation of flow problems with the mixed interface-tracking/interface-capturing technique (MITICT). *Computers & Fluids* 2007; **36**:2–11.
25. Formaggia L, Miglio E, Mola A, Parolini N. Fluid–structure interaction problems in free surface flows: application to boat dynamics. *International Journal for Numerical Methods in Fluids* 2008; **56**:965–978.
26. Martin D, Löhner R. An implicit linelet-based solver for incompressible flows. *AIAA-92-0668*, 1992.
27. Ramamurti R, Löhner R. A parallel implicit incompressible flow solver using unstructured meshes. *Computers & Fluids* 1996; **5**:119–132.
28. Sharov D, Luo H, Baum JD, Löhner R. Implementation of unstructured grid GMRES+LU-SGS method on shared-memory, cache-based parallel computers. *AIAA-00-0927* 2000.
29. Soto O, Löhner R, Cebral JR. An implicit monolithic time accurate finite element scheme for incompressible flow problems. *AIAA-01-2616-CP* 2001.
30. Soto O, Löhner R, Camelli F. A linelet preconditioner for incompressible flow solvers. *International Journal for Heat and Fluid Flow* 2003; **13**:133–147.
31. Löhner R. Multistage explicit advective prediction for projection-type incompressible flow solvers. *Journal of Computational Physics* 2004; **195**:143–152.

32. Löhner R, Yang C, Cebral JR, Camelli F, Soto O, Waltz J. Improving the speed and accuracy of projection-type incompressible flow solvers. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:3087–3109.
33. Tilch R, Tabbal A, Zhu M, Decker F, Löhner R. Combination of body-fitted and embedded grids for external vehicle aerodynamics. *Engineering Computations* 2008; **25**:28–41.
34. Cruchaga MA, Celentano DJ, Tezduyar TE. Moving-interface computations with the edge-tracked interface locator technique (ETILT). *International Journal for Numerical Methods in Fluids* 2005; **47**:451–469.
35. ten Cate A, Nieuwstad CH, Derksen JJ, Van den Akker HEA. Particle imaging velocimetry experiments and lattice-Boltzmann simulations on a single sphere settling under gravity. *Physics of Fluids* 2002; **14**:4012–4025.
36. Johnson AA, Tezduyar TE. Simulation of multiple spheres falling in a liquid-filled tube. *Computer Methods in Applied Mechanics and Engineering* 1996; **134**:351–373.
37. Johnson AA, Tezduyar TE. 3D simulation of fluid–particle interactions with the number of particles reaching 100. *Computer Methods in Applied Mechanics and Engineering* 1997; **145**:301–321.
38. Johnson AA, Tezduyar TE. Advanced mesh generation and update methods for 3D flow simulations. *Computational Mechanics* 1999; **23**:130–143.
39. Johnson AA, Tezduyar TE. Methods for 3D computation of fluid-object interactions in spatially-periodic flows. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:3201–3221.
40. Laverty S M Jr. Experimental hydrodynamics of spherical projectiles impacting on a free surface using high speed imaging techniques. *MSc Thesis*, Ocean Engineering, MIT, 2004.
41. Kim J, Moin P. Application of a fractional-step method to incompressible Navier–Stokes equations. *Journal of Computational Physics* 1985; **59**:308–323.
42. Bell JB, Colella P, Glaz H. A second-order projection method for the Navier–Stokes equations. *Journal of Computational Physics* 1989; **85**:257–283.
43. Bell JB, Marcus DL. A second-order projection method for variable-density flows. *Journal of Computational Physics* 1992; **101**(2):334–348.
44. Alessandrini B, Delhommeau G. A multigrid velocity-pressure-free surface elevation fully coupled solver for calculation of turbulent incompressible flow around a hull. *Proceedings of 21st Symposium on Naval Hydrodynamics*, Trondheim, Norway, 1996.
45. Kallinderis Y, Chen A. An incompressible 3-D Navier–Stokes method with adaptive hybrid grids. *AIAA-96-0293*, 1996.
46. Gresho PM, Upson CD, Chan ST, Lee RL. Recent progress in the solution of the time-dependent, three-dimensional, incompressible Navier–Stokes equations. In *Proceedings of the 4th International Symposium on Finite Element Methods in Flow Problems*, Kawai T (ed.). University of Tokyo Press, 1982.
47. Donea J, Giuliani S, Laval H, Quartapelle L. Solution of the unsteady Navier–Stokes equations by a fractional step method. *Computer Methods in Applied Mechanics and Engineering* 1982; **30**:53–73.
48. Gresho PM, Chan ST. On the theory of semi-implicit projection methods for viscous incompressible flows and its implementation via a finite element method that also introduces a nearly-consistent mass matrix. Part 2: implementation. *International Journal for Numerical Methods in Fluids* 1990; **11**:621–659.
49. Takamura A, Zhu M, Vinteler D. Numerical Simulation of Pass-by Maneuver Using ALE Technique. *JSAE Annual Conference*, Spring, Tokyo, 2001.
50. Eaton E. Aero-acoustics in an Automotive HVAC Module. *American PAM User Conference*, Birmingham, Michigan, 2001.
51. Karbon KJ, Kumarasamy S. Computational aeroacoustics in automotive design, computational fluid and solid mechanics. *First MIT Conference on Computational Fluid and Solid Mechanics*, Boston, 2001; 871–875.
52. Nichols BD, Hirt CW. Methods for calculating multi-dimensional, transient free surface flows past bodies. *First International Conference on Numerical Ship Hydrodynamics*, Gaithersburg, ML, 1975; 20–23.
53. Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**:201–225.
54. Scardovelli R, Zaleski S. Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics* 1999; **31**:567–603.
55. Fekken G, Veldman AEP, Buchner B. Simulation of green water loading using the Navier–Stokes equations. *7th International Conference on Numerical Ship Hydrodynamics*, Nantes, France, 1999.
56. Biausser BP, Fraunie P, Grilli S, Marcer R. Numerical analysis of the internal kinematics and dynamics of three-dimensional breaking waves on slopes. *International Journal of Offshore and Polar Engineering* 2004; **14**:247–256.
57. Sussman M, Smereka P, Osher S. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics* 1994; **114**:146–159.
58. Enright D, Nguyen D, Gibou F, Fedkiw R. Using the particle level set method and a second order accurate pressure boundary condition for free surface flows. In *4th ASME-JSME Joint Fluids Engineering Conference*, Kawahashi M, Ogut A, Tsuji Y (eds): Honolulu, HI, 2003.
59. Sussman M, Puckett E. A coupled level set and volume of fluid method for computing 3D and axisymmetric incompressible two-phase flows. *Journal of Computational Physics* 2000; **162**:301–337.
60. Osher S, Fedkiw R. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag: New York, 2003.
61. Sethian JA. *Level Set Methods and Fast Marching Methods*. Cambridge University Press: Cambridge, 1999.

DOI: 10.1002/fld

62. Tezduyar T, Aliabadi S, Behr M. Enhanced-discretization interface-capturing technique (EDICT) for computation of unsteady flows with interfaces. *Computer Methods in Applied Mechanics and Engineering* 1998; **155**:235–248.
63. Löhner R. Some useful renumbering strategies for unstructured grids. *International Journal for Numerical Methods in Engineering* 1993; **36**:3259–3270.
64. Löhner R. Solvers operating on shared-memory, cache-based parallel machines. *Computer Methods in Applied Mechanics and Engineering* 1998; **163**:95–109.
65. Lesoinne M, Farhat C. Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. *Computer Methods in Applied Mechanics and Engineering* 1996; **134**:71–90.
66. Cruchaga MA, Oñate E. A finite element formulation for incompressible flow problems using a generalized streamline operator. *Computer Methods in Applied Mechanics and Engineering* 1997; **143**:49–67.
67. Cruchaga MA, Oñate E. A generalized streamline finite element approach for the analysis of incompressible flow problems including moving surfaces. *Computer Methods in Applied Mechanics and Engineering* 1999; **173**:241–255.
68. Cruchaga MA, Celentano DJ, Tezduyar TE. A moving Lagrangian interface technique for flow computations over fixed meshes. *Computer Methods in Applied Mechanics and Engineering* 2001; **191**:525–543.
69. Cruchaga MA, Celentano DJ, Tezduyar TE. Computation of mould filling processes with a moving Lagrangian interface technique. *Communications in Numerical Methods in Engineering* 2002; **18**:483–493.
70. Tezduyar TE, Mittal S, Ray SE, Shih R. Incompressible flow computations with stabilized bilinear and linear equal-order interpolation velocity-pressure elements. *Computer Methods in Applied Mechanics and Engineering* 1992; **95**:221–242.
71. Brooks AN, Hughes TJR. Streamline upwind/Petrov Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1982; **32**:199–259.
72. Tezduyar TE. Finite elements methods for fluid dynamics with moving boundaries and interfaces. In *Encyclopedia of Computational Mechanics*, Vol. 3, Stein E, De Borst R, Hughes TJR (eds), Chapter 17. John Wiley & Sons, 2004.
73. Löhner R. Three-dimensional fluid–structure interaction using a finite element solver and adaptive remeshing. *Computer Systems in Engineering* 1990; **1**:257–272.

DOI: 10.1002/fld