Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 208 (2005) 289–314

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

An arbitrary Lagrangian Eulerian method for moving-boundary problems and its application to jumping over water

Jie Li [a,b,*], Marc Hesse [a], Johanna Ziegler [a], Andrew W. Woods [a]

[a] *BP Institute, University of Cambridge, Cambridge, UK*
[b] *Engineering Department, University of Cambridge, Cambridge, UK*

Received 2 September 2004; received in revised form 16 February 2005; accepted 17 February 2005
Available online 14 April 2005

Abstract

We develop an ALE (Arbitrary Lagrangian Eulerian) moving mesh method suitable for solving two-dimensional and axisymmetric moving-boundary problems, including the interaction between a free-surface and a solid structure. This method employs a body-fitted grid system where the gas–liquid interface and solid–liquid interface are lines of the grid system, and complicated dynamic boundary conditions are incorporated naturally and accurately in a Finite-Volume formulation. The resulting nonlinear system of mass and momentum conservation is then solved by a fractional step (projection) method. The method is validated on the uniform flow passing a cylinder (a two-dimensional flow with a solid structure) and several problems of bubble dynamics (axi-symmetrical flows with a free surface) for both steady and unsteady flows. Good agreement with other theoretical, numerical and experimental results is obtained. A further application is the investigation of a two-dimensional mechanical strider (a mass-spring system) interacting with a water surface, demonstrating the ability of the method in handling the interaction between a solid structure and a free surface. We find that the critical compression required to jump off the water surface varies linearly with spring constant for stiff springs and algebraically with exponent 0.7 for weak springs.

Keywords: ALE; Moving boundary; Free-surface; Finite volume method; Moving mesh method; Projection method; Multigrid method

[*] Corresponding author. Tel.: +1223 765707; fax: +1223 765701.
E-mail address: jie@bpi.cam.ac.uk (J. Li).

doi:10.1016/j.jcp.2005.02.016

1. Introduction

In recent years there has been a growing interest in bio-mechanical flows, which often involve solid structures and free surfaces as well as the interaction between them. One example is the hydrodynamics underlying the surface locomotion of semi-aquatic insects [1]. Fig. 1 shows a leg of a static strider resting on the free surface. The distortion of this surface generates a curvature force per unit leg length equal to $2\gamma\sin(\theta)$, where γ is the surface tension of water, and θ the contact angle with the horizontal. Walking or jumping of the water strider on the liquid surface is a highly nonlinear process, involving the interaction between a moving free surface and a solid structure. One of the greatest difficulties is that the location and the shape of the air–liquid–structure interfaces (boundaries) are time-dependent. During the flow these boundaries evolve, and may undergo severe deformations. These moving boundaries play a major role in defining the system and must be determined as part of the solution. In a numerical treatment of the moving boundary, we ought to address three questions: (1) how do we represent the boundary on a finite mesh? (2) how will the boundary evolve in time? and (3) how should we apply boundary conditions on the boundary? The aim of this work is to develop a novel numerical method suitable for these moving boundary problems.

Numerical solutions of moving boundary problems can be classified into two categories, moving grid methods and the fixed grid methods. Each method has its own advantages and disadvantages. In the context of fixed grid methods, there are surface tracking methods and volume tracking methods, such as the front-tracking method [2,3], the volume of fluid (VOF) method [4–7] and the level-set method [8]. Methods in this category are capable of simulating very complex interface motion, and new algorithms have been developed to improve the accuracy in the treatment of boundary conditions [9,10]. Moving grid methods employ a so-called boundary-fitted grid system [11–13]. This type of method has the tremendous advantage that the boundary condition is treated neatly and resolved very accurately, because the moving boundary coincides with one line of the numerical grid. The numerical method developed in this work falls into this second category.

We will discuss our ALE method on a stationary grid, before we extend it to a moving grid. Numerical methods for incompressible fluid flows have been successfully developed for fixed grids with curvilinear coordinates. One of the popular methods is the fractional step or projection method using the primitive variables velocity and pressure as computational variables [14]. This method decouples the continuity equation from the momentum equations by solving a Poisson-like elliptic equation for the pressure. Rosenfeld et al. [15] employed the volume fluxes defined on cell faces as primary variables and discretized the governing equations on a staggered grid using a finite-volume method (Fig. 2(a)). This approach is a natural

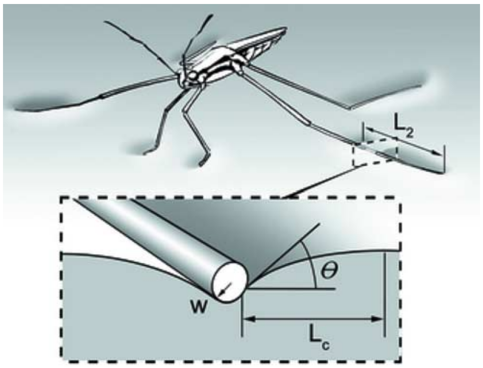


Fig. 1. A leg of a static strider on the free surface, whose distortion generates the curvature force per unit leg length $2\gamma\sin(\theta)$ that supports the strider's weight (Courtesy Hu et al. [1]).

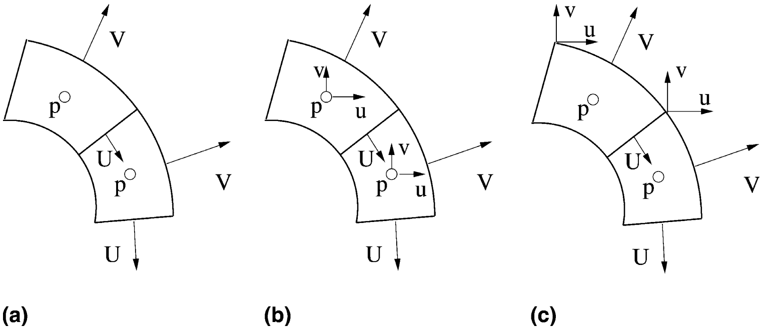


Fig. 2. Three types of meshes for numerical methods in curvilinear coordinates: (a) a staggered grid with the volume flux defined on its corresponding cell face; (b) a colocated-grid method with additional velocity components defined at cell center; (c) a ALE method with additional velocity components defined at cell corner.

extension of the standard staggered-grid method (also call MAC method) [16] in Cartesian coordinates. The discretization of the momentum equations is quite complicated, and the numerical computation of these equations expensive. The use of Cartesian velocity components as the primary dependent variables simplifies the form of the governing equations and their discretization is straightforward. However, solution methods for the incompressible flow, based on a non-staggered grid in the classic form, are not stable and produce spurious oscillations in the pressure equation, i.e., the "checkerboard" pattern [17]. In order to overcome this difficulty, Rhie and Chow's method [18] combined the stability of the staggered grid for the pressure equation and the efficiency of the non-staggered grid in computing the momentum equations. They computed the velocity components in the momentum equations at the cell center, and reinforced the mass conservation (the continuity equation) on the staggered grid (Fig. 2(b)). The volume fluxes U and V needed in the pressure equation are interpolated from the velocity u and v defined at the cell center. The use of the pressure gradient in the momentum equation and the pressure equation is not consistent in this method. The mass conservation is only satisfied on second order accuracy [19]. This method has been applied to unsteady flow problems [20,21].

In this work, we will employ a semi-staggered grid method, which is a special case of the more general ALE (Arbitrary Lagrangian Eulerian) method. Our method is a compromise between the two methods mentioned above: although the velocity components are defined at the same location (at the cell corner), their location is different from the pressure location at the center (Fig. 2(c)). This grid layout has received less attention in the finite-volume community. It was, however, widely advocated by the finite-element community who named it the Q_1Q_0 method. A very attractive feature of this method, shown in [22], is that the dynamic boundary condition can be incorporated naturally and accurately. This is the reason we have chosen the ALE method as the basis of this work. However, our ALE method is different from the Q_1Q_0 method used in the finite-element community. First we use a finite-volume methodology to discretise the pressure equation and momentum equations. More importantly, the Q_1Q_0 method is marginally unstable because of the checkerboard oscillation pattern in the pressure equation. The treatment of the pressure in our ALE method is similar to the one of the colocated-grid method we discussed above. The mass conservation is reinforced on the pressure cell in our method, and the volume fluxes U and V needed in the pressure equation are interpolated from the velocity u and v at the cell corners. In summary, the staggered-grid method, the colocated-grid method and the ALE method share the common approach in dealing with the pressure equation, and its solution is efficient. Indeed, the same pressure solver can be used in these three methods without any change once the right-hand side (the source term) is computed.

For an initial-value problem with moving boundaries, a numerical method must address a crucial question: at which time level the boundary conditions must be satisfied. For reasons of simplicity and clarity, we employ an explicit scheme in our method. We suppose that we know the velocity at time level n. All boundary conditions are to be satisfied on time level n, and the velocity field at the time level $n + 1$ is computed on the grid of time level n. We then generate a new grid for time level $n + 1$, and interpolate the velocity to the new grid. Our method is therefore first order in time. This first order accuracy is not a constraint for the proposed investigation of a mechanical strider (a mass-spring system) interacting with a water surface in this work. We note that our method is not a finite-volume method in the strict sense. The finite-volume methodology is applied only to the stress tensor term while non-conservative schemes are used to treat advective terms [4]. This is consistent with the use of the above interpolation, which does not conserve the physical quantities neither. We conceived a non-conservative method because of its simplicity. In many cases, a conservative method may be desirable. For example, based on discretizing time as well as space with the finite volume principle, the integrated space–time finite-volume method enforces global conservation for time-dependent moving boundary problems [23].

Bio-mechanical flows often involve non-Newtonian liquids with surfactant affecting the surface tension of the air–fluid interface. For example, some insects drift on water by injecting a chemical at the rear. This chemical reduces the surface tension behind their bodies so that the insects are pulled forward. Another example is the airway reopening flow, where the air has to enter the fluid-filled airways in the lung of a newborn baby for the first time. In this work, we consider only a Newtonian liquid with a constant surface tension. It is, however, easy to incorporate properties of a non-Newtonian liquid and a surfactant induced variation on the surface tension in our method. This feature makes the ALE framework very attractive.

2. Governing equations and boundary conditions

The fluid flows involving free-surfaces, solid structures as well as the interaction between them are often highly nonlinear processes. We shall compute these flows numerically. The flow of the liquid is governed by the incompressible Navier–Stokes equations. In the two-dimensional case, the continuity equation reads

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0, \tag{1}$$

and the momentum equation

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{\partial p}{\partial x}+\frac{\partial}{\partial x}\left(2\mu\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\right), \tag{2}$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=-\frac{\partial p}{\partial y}+\frac{\partial}{\partial x}\left(\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\right)+\frac{\partial}{\partial y}\left(2\mu\frac{\partial v}{\partial y}\right)+g, \tag{3}$$

where u and v are horizontal and vertical velocities, and p is the pressure. We consider the gas to be incompressible and at constant pressure p_0. Without loss of generality, we assume the constant pressure $p_0 = 0$. Hence the dynamic boundary condition on the free-surface expresses the following force balance:

$$(-p\mathbf{I}+\mu\mathbf{D})\cdot\mathbf{n}=\gamma\kappa\mathbf{n}, \tag{4}$$

where

$$\mathbf{D}=\left[\nabla\mathbf{u}+(\nabla\mathbf{u})^{\mathrm{T}}\right]=\begin{pmatrix}2u_x & u_y+v_x\\ u_y+v_x & 2v_y\end{pmatrix} \tag{5}$$

is the strain rate tensor, γ the surface tension, κ the curvature, and $\mathbf{n}$ the normal vector of interface. And we also require the kinematic boundary condition on the gas–liquid interface:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{u}, \quad \text{where } \mathbf{x} = (x, y), \tag{6}$$

which means the interface moves with the speed of the fluid.

A well-studied boundary condition on the structure–liquid interface is a Dirichlet condition on the velocity. We have also the possibility to prescribe a force on this interface as boundary condition. This type of boundary condition is in practice satisfied by iterating on the Dirichlet condition of the velocity. We should discuss this boundary condition in detail later.

3. Numerical method

We employ a moving grid method in which the free surface and the structure–fluid interface coincide with lines of the numerical grid. It is an appropriate choice for the flow problem we study in this work, since the interface only undergoes moderate deformation. For the case where the interface undergoes severe deformation, even breakup, a fixed mesh method is more appropriate. Condition on the free surface is expressed naturally in terms of the primitive variables, see Eq. (4), this fact motivates us to develop a new ALE (Arbitrary Lagrangian Eulerian) method, where the spatial discretization of the primitive variables uses a partially staggered approach: the pressure p is defined at the cell center while both components of the Cartesian velocity $\mathbf{u}$ are defined on the cell corner (see Fig. 3). As shown in [22], the dynamic boundary condition can be incorporated naturally and accurately in the ALE method. This is done by expressing the stress force in the momentum equation in the finite-volume formulation. For a cell on the boundary with area A_Ω and face lengths Δl (see Fig. 3),

$$\begin{aligned}\nabla \cdot (p\mathbf{I} + \mu\mathbf{D})A_\Omega &\approx \int_\Omega \nabla \cdot (p\mathbf{I} + \mu\mathbf{D})\mathrm{d}A = \int_{\partial\Omega} (p\mathbf{I} + \mu\mathbf{D}) \cdot \mathbf{n}\,\mathrm{d}l \\ &\approx [(p\mathbf{I} + \mu\mathbf{D}) \cdot \mathbf{n}\Delta l)]_\mathrm{E} + [(p\mathbf{I} + \mu\mathbf{D}) \cdot \mathbf{n}\Delta l)]_\mathrm{W} + [(p\mathbf{I} + \mu\mathbf{D}) \cdot \mathbf{n}\Delta l)]_\mathrm{N} + [(p\mathbf{I} + \mu\mathbf{D}) \cdot \mathbf{n}\Delta l]_\mathrm{S}.\end{aligned}$$

We apply the boundary condition on the south face. From Eq. (4), we substitute the stress force $(-p\mathbf{I} + \mu\mathbf{D})\mathbf{n}$ on the south face by the surface tension force $\gamma\kappa\mathbf{n}$. Our numerical method does not require the mesh to be orthogonal; the orthogonality of a mesh is desirable but not easy to obtain in many cases. Our method is therefore flexible.

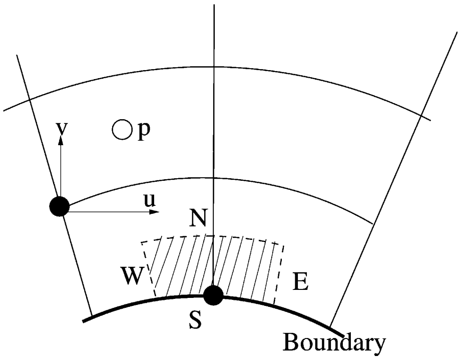


Fig. 3. ALE mesh. The pressure p is defined at the cell center while both Cartesian velocity u and v are defined on the cell corner.

3.1. Projection method

We first solve the coupled continuity equation (1) and momentum equations (2) and (3) with proper boundary conditions on a fixed mesh for the time $t^n = n\Delta t$. We employ an explicit projection method [14]. An approximate velocity $\mathbf{u}^*$ is calculated without the pressure gradient ∇p from the momentum equations. We assume that the velocity $\mathbf{u}^n$ at time t_n is known:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = \mathbf{F}(\mathbf{u}^n), \tag{7}$$

where the term $\mathbf{F}(\mathbf{u}^n)$ includes the convection, diffusion terms and body forces in the momentum equation. In general, the resulting flow field $\mathbf{u}^*$ does not satisfy the continuity equation. However, we require that $\nabla \cdot \mathbf{u}^{n+1} = 0$ and

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

Taking the divergence of Eq. (8), we obtain a Poisson-like equation

$$\nabla \cdot (\nabla p^n) = \frac{\nabla \cdot \mathbf{u}^*}{\Delta t} = \nabla \cdot (\mathbf{F}(\mathbf{u}^n)). \tag{9}$$

The solution of this equation is determined with appropriate boundary conditions for the pressure.

The velocity and the pressure are intimately related in an incompressible flow. Only one of them can be prescribed on a boundary. A commonly used boundary condition is a Dirichlet condition on the velocity. This is also the simplest boundary condition to use on a fluid–structure interface. The relevant boundary condition on the pressure is a Neumann-type condition. On the free interface, the boundary condition for the pressure is dictated by the balance of the normal stresses

$$p^n = \mathbf{n}(\mu \mathbf{D}^n)\mathbf{n} - \gamma^n \kappa^n. \tag{10}$$

This condition is derived from Eq. (4) and is a Dirichlet condition for the pressure. As pointed out in [22], the time 'index' on p in Eqs. (8) and (9) is perfectly proper; it should not be p^{n+1} as many believe. Eq. (9) is the best equation to use to prove this assertion. The RHS of Eq. (9) and the Dirichlet boundary condition Eq. (10) are completely determined by terms at the time step n, and we should not forget that the mesh we use here is still the one at the time step n.

3.2. Moving mesh method

We remind the reader that until now we have worked on the mesh at time t^n. From Eq. (8), we compute the velocity field for the time t^{n+1} on this same mesh. We then compute the new free surface and the fluid–structure interface through the Lagrangian formula

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \mathbf{u}(\Delta t). \tag{11}$$

In our method, the interface is represented by a series of marker points $\mathbf{r}_i$, $i = 1, \ldots, N$, on the surface. A cubic spline is used to obtain a continuous smooth parametrization of the surface and compute the surface curvature accurately.

Finally, we generate a new mesh $\mathbf{x}^{n+1}$ which fits the new free surface and the new fluid–structure interface. The velocity field on the new mesh is extrapolated through

$$\mathbf{u}^{n+1}(\mathbf{x}^{n+1}) = \mathbf{u}^{n+1}(\mathbf{x}^n) + (\mathbf{x}^{n+1} - \mathbf{x}^n) \cdot \nabla \cdot (\mathbf{u}^{n+1})(\mathbf{x}^n). \tag{12}$$

This step can also be interpreted from the composite rule of derivatives

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \frac{\partial \mathbf{u}}{\partial t} + \mathbf{v}\nabla\mathbf{u}, \tag{13}$$

where $\mathbf{v}$ is the mesh motion speed.

3.3. Spatial discretizations

The above subsection discusses the temporal discretization of the governing equation. Now we address the spatial discretization. We first provide the mathematical formulae related to the curvilinear coordinates that we need in the discretization of the momentum and the pressure equations. We follow the development given by [24,21] and the geometric arguments in [25]. Given a set of curvilinear coordinates ξ and η, with unit spacing on the computational grid in the logical domain ($\Delta\xi = \Delta\eta = 1$), integers (i, j) can be used as discrete curvilinear coordinates. Fig. 4 shows a typical mesh cell in the physical domain (x, y) and in the computational plane (i, j). Note that the mesh is defined by two-dimensional grid points $(\hat{x}_{i,j}, \hat{y}_{i,j})$, while the center $(x_{i,j}, y_{i,j})$ of cell (i, j) is defined as the average of the coordinates of its four corners. We denote the physical quantity ϕ at the cell center as $\phi_{i,j}$, at the right face center $\phi_{i+\frac{1}{2},j}$, and at the top face center $\phi_{i,j+\frac{1}{2}}$.

All Cartesian derivatives of a variable, i.e. ϕ, can be related to the ξ and η derivatives using the chain rule,

$$\phi_\xi = \phi_x x_\xi + \phi_y y_\xi,$$
$$\phi_\eta = \phi_x x_\eta + \phi_y y_\eta,$$

and solving the linear system for ϕ_x and ϕ_y, we obtain

$$\phi_x = \frac{1}{J}\left(y_\eta\phi_\xi - y_\xi\phi_\eta\right), \tag{14}$$

$$\phi_y = \frac{1}{J}\left(x_\xi\phi_\eta - x_\eta\phi_\xi\right), \tag{15}$$

$$J = x_\xi y_\eta - x_\eta y_\xi. \tag{16}$$

J is called the Jacobian of the transformation from physical space with Cartesian coordinates x and y to logical space with Cartesian coordinates ξ and η.

As reported in [4], a conservative scheme loses, in general, one order of formal precision for the advective terms. Hence, we chose a non-conservative finite-difference scheme for these terms. All derivatives in the

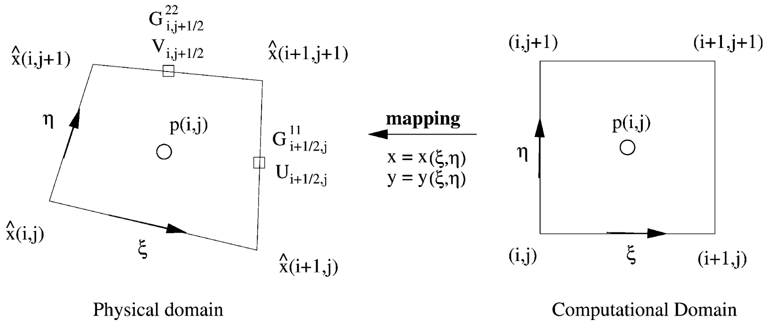


Fig. 4. The same computational mesh shown in the physical and the logical domain.

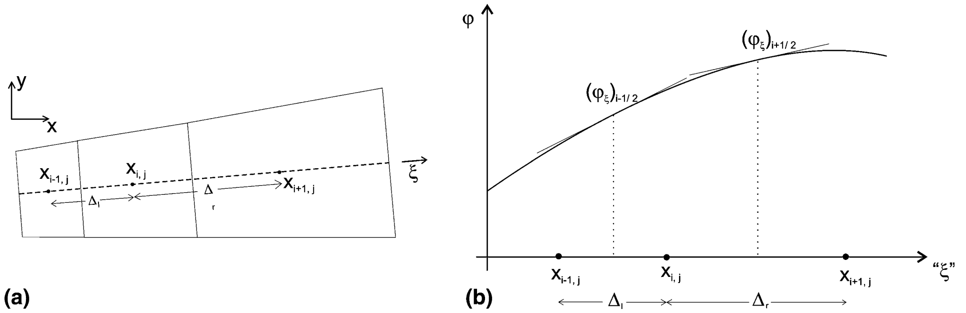


Fig. 5. (a) Three velocity cells and nodes along a line of $\eta = \eta_0$ and increasing ξ on a non-uniform mesh. (b) The interpolation scheme used to obtain the first derivatives with respect to the logical coordinates.

velocity gradient $\nabla\mathbf{u}$ are mapped into the logical domain using Eqs. (14) and (15). The derivatives in the logical domain are approximated using the interpolation scheme shown in Fig. 5(b).

$$\left(\frac{\partial\phi}{\partial\xi}\right)_{i,j} = \lambda\left(\frac{\partial\phi}{\partial\xi}\right)_{i-1/2,j} + (1-\lambda)\left(\frac{\partial\phi}{\partial\xi}\right)_{i+1/2,j}, \tag{17}$$

where the derivatives at the midpoints $(\phi_\xi)_{i+1/2}$ and $(\phi_\xi)_{i-1/2}$ are approximated by central differences, and $\lambda = \Delta_r/(\Delta_l + \Delta_r)$ with $\Delta_l = \|\mathbf{x}_{i,j} - \mathbf{x}_{i-1,j}\|$ and $\Delta_r = \|\mathbf{x}_{i+1,j} - x_{i,j}\|$. Derivatives with respect to η are approximated in a similar way. Given that the velocity nodes are uniformly spaced in logical space, a standard central difference is appropriate. The basic idea in Eq. (17) is to weight the derivatives by cell size such that the correct order of approximation is maintained in a curvilinear mesh. On a uniform Cartesian mesh, the scheme reduces to a central difference.

The pressure and viscous terms in the momentum equation are discretized using the finite-volume formulation. Consider a velocity cell (see Fig. 6),

$$\nabla\cdot\mathbf{D} \approx \frac{1}{A_\Omega}\int_\Omega \nabla\cdot\mathbf{D}\,\mathrm{d}A = \frac{1}{A_\Omega}\int_{\partial\Omega}\mathbf{D}\cdot\mathbf{n}\,\mathrm{d}l \approx \frac{1}{A_\Omega}\left[(\mathbf{D}\cdot\mathbf{n}\Delta l)_\mathrm{E} + (\mathbf{D}\cdot\mathbf{n}\Delta l)_\mathrm{W} + (\mathbf{D}\cdot\mathbf{n}\Delta l)_\mathrm{N} + (\mathbf{D}\cdot\mathbf{n}\Delta l)_\mathrm{S}\right],$$

where A_Ω is the cell area and Δl face lengths. The derivatives in the tensor $\mathbf{D}$ on the face centers are computed by finite difference through Eqs. (14) and (15). The viscous terms are treated explicitly in our method.

The same finite-volume methodology is applied to the pressure equation (9) on the pressure cells. This is very well explained in [24], and the final result is expressed in curvilinear coordinates, as

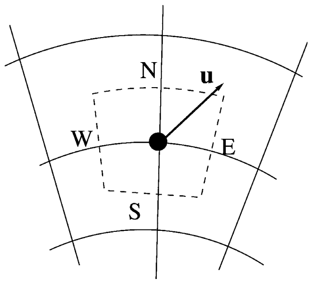


Fig. 6. ALE mesh. The pressure p is defined at the cell center while both Cartesian velocity u and v are defined on the cell corner.

$$\left(G^{11}p_\xi\right)_{i+1/2,j} - \left(G^{12}p_\eta\right)_{i+1/2,j} - \left(G^{11}p_\xi\right)_{i-1/2,j} + \left(G^{12}p_\eta\right)_{i-1/2,j} - \left(G^{21}p_\xi\right)_{i,j+1/2} + \left(G^{22}p_\eta\right)_{i,j+1/2} + \left(G^{21}p_\xi\right)_{i,j-1/2} - \left(G^{22}p_\eta\right)_{i,j-1/2} = \mathrm{RHS}, \tag{18}$$

where

$$G^{11}_{i+\frac{1}{2},j} = J^{-1}\left(\hat{x}_\eta x_\eta + \hat{y}_\eta y_\eta\right)_{i+\frac{1}{2},j},$$
$$G^{12}_{i+\frac{1}{2},j} = J^{-1}\left(\hat{x}_\eta x_\xi + \hat{y}_\eta y_\xi\right)_{i+\frac{1}{2},j},$$
$$G^{21}_{i,j+\frac{1}{2}} = J^{-1}\left(x_\eta \hat{x}_\xi + y_\eta \hat{y}_\xi\right)_{i,j+\frac{1}{2}},$$
$$G^{22}_{i,j+\frac{1}{2}} = J^{-1}\left(x_\xi \hat{x}_\xi + y_\xi \hat{y}_\xi\right)_{i,j+\frac{1}{2}}$$

are Christoffel symbols which measure the degree of deformation of a coordinate system. This is a well-investigated system as far as the interior points are concerned. There is a subtlety in the numerical approximations of the Christoffel symbols [24]. The hat (ˆ) there indicates that the derivatives are to be approximated with two-points finite difference,

$$\left(\hat{x}_\eta\right)_{i-\frac{1}{2},j} = \hat{x}_{i,j+1} - \hat{x}_{i,j},$$
$$\left(\hat{x}_\xi\right)_{i,j-\frac{1}{2}} = \hat{x}_{i+1,j} - \hat{x}_{i,j},$$

while the derivatives without (ˆ) are to be approximated with four-points finite difference,

$$\left(x_\eta\right)_{i-\frac{1}{2},j} = \frac{1}{4}\left(x_{i,j+1} - x_{i,j-1} + x_{i-1,j+1} - x_{i-1,j-1}\right),$$
$$\left(x_\xi\right)_{i,j-\frac{1}{2}} = \frac{1}{4}\left(x_{i+1,j} - x_{i-1,j} + x_{i+1,j-1} - x_{i-1,j-1}\right).$$

Similar approximations are used for derivatives of $\hat{y}$ and y.

The novelty of our pressure discretization is the treatment of a mixed boundary condition. In Fig. 7, the thick solid represents a structure–fluid interface (a Neumann condition) and the thick dashed line a

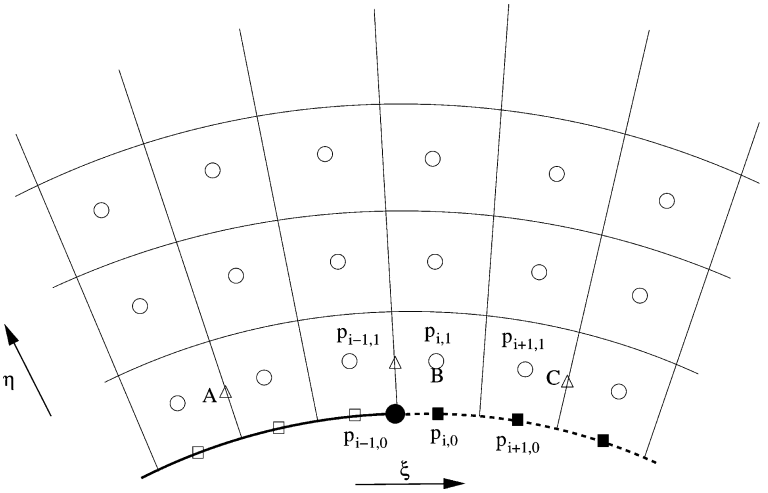


Fig. 7. Mixed boundary condition for the pressure. The thick solid represents a solid wall (a Neumann condition) and the thick dashed line a free-surface (a Dirichlet condition). The black circle is the transition point. The Dirichlet boundary condition is imposed at black squares and the Neumann condition at blank squares on the boundary.

free-surface (a Dirichlet condition). The black circle is the transition point. The Dirichlet boundary condition is imposed at black squares and the Neumann condition at blank squares on the boundary. The pressure to be computed is defined at the cell center, represented by blank circles. The Neumann condition was addressed by previous authors [24], and is incorporated naturally in Eq. (18). There is no need to compute the pressure derivatives on the Neumann boundary. However, the derivative should be estimated at off-boundary points. One such point is indicated by A in Fig. 7. There is some ambiguity on how this derivative should be computed, and one-sided approximations next to the boundaries

$$(p_\eta)_{i+\frac{1}{2},1} = \frac{1}{2}(p_{i,2} - p_{i,1} + p_{i+1,2} - p_{i+1,1}).$$

must be used. On the free-surface, the pressure is given by Eq. (10). We use the simplest finite differences to approximate the pressure derivatives on the free-surface:

$$\begin{aligned}
&(p_\eta)_{i+\frac{1}{2},1} = p_{i,1} - p_{i,0} + p_{i+1,1} - p_{i+1,0},\\
&(p_\eta)_{i,\frac{1}{2}} = 2(p_{i,1} - p_{i,0}),\\
&(p_\xi)_{i,\frac{1}{2}} = \frac{1}{2}(p_{i+1,0} - p_{i-1,0}),
\end{aligned}$$

with one exception at the transition cell indicated by B where $p_{i-1,0}$ is on the fluid-structure interface and is unknown. We use a one sided finite-difference for simplicity,

$$(p_\xi)_{i,\frac{1}{2}} = (p_{i+1,0} - p_{i-1,0}).$$

The solution of the discrete Poisson's equation (18) is the most time-consuming part of our Navier–Stokes solver and, consequently, an efficient solution is crucial for the performance of the whole method. The multigrid method is arguably the most efficient method: to reduce the error in the discretization of Poisson's equation by a constant, the multigrid method requires a fixed number of iterations, independent of the mesh size. Our multigrid method was adapted from the one developed in [21, Chapter 6]. The trick is to transform a system with non-homogeneous boundary conditions into a system with homogeneous boundary conditions. In the later system, all terms related to the boundary conditions are moved to the right-hand side, so that the system is assembled in a unified form for both interior and boundary cells as

$$\begin{aligned}
&C^{ne}_{ij}p_{i+1,j+1} + C^{e}_{ij}p_{i+1,j} + C^{se}_{ij}p_{i+1,j-1} + C^{n}_{ij}p_{i,j+1} + C^{c}_{ij}p_{ij} + C^{s}_{ij}p_{i,j-1} + C^{nw}_{ij}p_{i-1,j+1}\\
&\quad + C^{w}_{ij}p_{ij} + C^{sw}_{ij}p_{i-1,j-1} = \mathrm{RHS},
\end{aligned} \tag{19}$$

where C^{*}_{ij} are nine relevant coefficients related to cell (i, j). These coefficients are zero when the relevant pressure terms are not defined inside the physical domain. The fact that the discrete pressure equation has the same form for both interior and boundary cells allows us to write out the Gauss–Seidel smoother in a simple and unified form:

$$\begin{aligned}
p^{m+1}_{ij} = \frac{1}{C^{c}_{ij}}(&\mathrm{RHS} - C^{ne}_{ij}p^{m}_{i+1,j+1} - C^{e}_{ij}p^{m}_{i+1,j} - C^{se}_{ij}p^{m+1}_{i+1,j-1} - C^{n}_{ij}p^{m}_{i,j+1} - C^{s}_{ij}p^{m+1}_{i,j-1} - C^{nw}_{ij}p^{m}_{i-1,j+1}\\
&- C^{w}_{ij}p^{m+1}_{i-1,j} - C^{sw}_{ij}p^{m+1}_{i-1,j-1}),
\end{aligned}$$

where m is the number of the Gauss–Seidel iterations. We generally observe that our multigrid method converges slightly faster with the mixed boundary condition than the pure Neumann condition.

We show how the assemblage of the coefficients is done for the transition cell. This is the most complicated case illustrating the originality of our method. For the cell indicated by B in Fig. 7, the derivatives on the east face are approximated as

$$
\begin{aligned}
(p_\xi)_{i+\frac{1}{2},1} &= p_{i+1,1} - p_{i,1},\\
(p_\eta)_{i+\frac{1}{2},1} &= p_{i+1,1} + p_{i,1} - p_{i+1,0} - p_{i,0}.
\end{aligned}
$$

On the west face,

$$
\begin{aligned}
(p_\xi)_{i-\frac{1}{2},1} &= p_{i,1} - p_{i-1},\\
(p_\eta)_{i-\frac{1}{2},1} &= \frac{1}{2}(p_{i-1,2} + p_{i,2} - p_{i-1,1} - p_{i,1}),
\end{aligned}
$$

where we switch to the formula for Neumann condition as the pressure $p_{i-1,0}$ is on the fluid–structure interface and is not known. The north face is an ordinary interior face, where the derivatives are approximated by the conventional formula,

$$
\begin{aligned}
(p_\xi)_{i,\frac{3}{2}} &= \frac{1}{4}(p_{i+1,2} - p_{i-1,2} + p_{i+1,1} - p_{i-1,1}),\\
(p_\eta)_{i,\frac{3}{2}} &= p_{i,2} - p_{i,1}.
\end{aligned}
$$

And finally, on the south face,

$$
\begin{aligned}
(p_\eta)_{i,\frac{1}{2}} &= 2(p_{i,1} - p_{i,0}),\\
(p_\xi)_{i,\frac{1}{2}} &= p_{i+1,0} - p_{i,0},
\end{aligned}
$$

where we switch to the one-sided finite difference. The pressure equation for cell B is therefore

$$
\begin{aligned}
&G^{11}_{i+\frac{1}{2},1}(p_{i+1,1} - p_{i,1}) + G^{12}_{i+\frac{1}{2},1}(p_{i+1,1} + p_{i,1} - p_{i+1,0} - p_{i,0}) - G^{11}_{i-\frac{1}{2},1}(p_{i,1} - p_{i-1}) - \frac{1}{2}G^{12}_{i-\frac{1}{2},1}(p_{i-1,2} + p_{i,2} - p_{i-1,1}\\
&\quad - p_{i,1}) + \frac{1}{4}G^{21}_{i,\frac{3}{2}}(p_{i+1,2} - p_{i-1,2} + p_{i+1,1} - p_{i-1,1}) + G^{22}_{i,\frac{3}{2}}(p_{i,2} - p_{i,1}) - G^{21}_{i,\frac{1}{2}}(p_{i+1,0} - p_{i,0}) - 2G^{22}_{i,\frac{1}{2}}(p_{i,1} - p_{i,0})\\
&\quad = \text{RHS}.
\end{aligned}
$$

The pressure terms $p_{i+1,0}$ and $p_{i,0}$ on the boundary are known and are therefore moved to the right-hand side. On the other hand, the pressure $p_{i\text{-}1,0}$ on the solid boundary is not known but does not figure in the above equation. Hence $C^{se}_{i,1}$, $C^{s}_{i,1}$ and $C^{sw}_{i,1}$ are zero. The other coefficients are

$$
\begin{array}{lllllllll}
C^{ne}_{n,1} & = & & & & & -\frac{1}{4}G^{21}_{n,\frac{3}{2}}, & & \\
C^{n}_{n,1} & = & & & & +\frac{1}{2}G^{12}_{n-\frac{1}{2},1} & & +G^{22}_{n,\frac{3}{2}}, & \\
C^{nw}_{n,1} & = & & & & +\frac{1}{2}G^{12}_{n-\frac{1}{2},1} & +\frac{1}{4}G^{21}_{n,\frac{3}{2}}, & & \\
C^{e}_{n,1} & = & G^{11}_{n+\frac{1}{2},1} & -G^{12}_{n+\frac{1}{2},1} & & & -\frac{1}{4}G^{21}_{n,\frac{3}{2}}, & & \\
C^{c}_{n,1} & = & -G^{11}_{n+\frac{1}{2},1} & -G^{12}_{n+\frac{1}{2},1} & -G^{11}_{n-\frac{1}{2},1} & -\frac{1}{2}G^{12}_{n-\frac{1}{2},1} & & -G^{22}_{n,\frac{3}{2}} & -2G^{22}_{n,\frac{1}{2}}, \\
C^{w}_{n,1} & = & & & G^{11}_{n-\frac{1}{2},1} & -\frac{1}{2}G^{12}_{n-\frac{1}{2},1} & +\frac{1}{4}G^{21}_{n,\frac{3}{2}}. & &
\end{array}
$$

3.4. Axisymmetric flows

We also investigate axisymmetric flows in this work. Under the assumption of axisymmetry, the continuity equation is

$$\frac{\partial u}{\partial z} + \frac{1}{r}\frac{\partial}{\partial r}(rv) = 0 \tag{20}$$

and the momentum equations

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial z} + v\frac{\partial u}{\partial r} = -\frac{\partial p}{\partial z} + \frac{\partial}{\partial z}\left(2\mu\frac{\partial u}{\partial z}\right) + \frac{1}{r}\frac{\partial}{\partial r}\left(r\mu\left(\frac{\partial u}{\partial r} + \frac{\partial v}{\partial z}\right)\right) + g, \tag{21}$$

$$\frac{\partial}{\partial t} + u\frac{\partial v}{\partial z} + v\frac{\partial v}{\partial r} = -\frac{\partial p}{\partial r} + \frac{\partial}{\partial z}\left(\mu\left(\frac{\partial u}{\partial r} + \frac{\partial v}{\partial z}\right)\right) + \frac{1}{r}\frac{\partial}{\partial r}(r(2\mu\frac{\partial v}{\partial r})) - \frac{2\mu v}{r^2}, \tag{22}$$

where u and v are the axial and radial velocities, and p the pressure. The 2D Cartesian finite-volume discretization can be extended to axially symmetric flows by changing the cell volumes and cell faces to axially symmetric geometry. The area of the cell faces is the length of the 2D cell side multiplied by the radius of the center of the side. The volume of an axisymmetric control volume is

$$\Omega = \frac{1}{6}\left[\sum_{i=1}^{3}(r_i + r_{i+1})(r_i z_{i+1} - r_{i+1} z_i) + (r_4 + r_1)(r_4 z_1 - r_1 z_4)\right], \tag{23}$$

where the subscripts 1–4 refer to the vertices labelled counter clockwise.

4. Validations

The new feature of our numerical method is the ability to resolve accurately the interaction between a free-surface and a solid structure. As a first step, we ensure that our method is accurate for two particular cases; flows with pure Dirichlet conditions for velocity (a solid structure) and flows with only free surfaces. We investigate the wake developing behind a cylinder in a uniform background flow. We validate our numerical method by comparing our numerical results to laboratory observations, and numerical results. We then investigate the dynamics of a bubble rising in a quiescent liquid. The unsteady oscillating bubble is also computed in order to compare with the theory of Prosperetti [26], and the normal-mode analysis. Finally, in Sections 5 and 6, the method is applied to a problem of interaction between a solid structure and a free-surface. We study a mass-spring system interacting with a water surface.

4.1. Flow past cylinder

The lid-driven flow in a two-dimensional polar cavity is a standard test case for the numerical methods written in curvilinear coordinates [15,20,21]. An interesting case is for Reynolds number $Re = 350$ (based on the lid velocity and the radius of the inner circle), where a discrepancy between the numerical and experimental data is observed and is attributed to the three-dimensional effects in experiments [27]. A more geometrically challenging flow is the uniform flow past a cylinder. This flow field depends strongly on the Reynolds number $Re = 2a\rho U/\mu$, where a is the radius of the cylinder, U is the velocity of the background flow, and μ is the dynamic viscosity. For $6 \leqslant Re < 40$, a steady state wake forms on the leeward side of the cylinder. This steady wake takes the form of two symmetric vortices [28]. This flow is used to check the spatial and temporal accuracy of our numerical method. To facilitate the comparison, we use similar numerical parameters as in [15]. The symmetric flow is solved using a non-orthogonal coordinate system with 48×80 mesh points in the radial and circumferential directions respectively. Exponential spacing of grid lines both in the radial and in the circumferential direction were used to cluster the points leeward of the cylinder. The flow domain is composed of two non-concentric half circles with the radius of the outer circle 40 times larger than that of the inner circle, and the center of the inner circle was displaced upstream by 10 units.

The time evolution of the separation length for *Re* = 40 is computed and shown in Fig. 8, together with well-known numerical and experimental results. The symbol × represents the results of this study, the triangle the experimental data of Coutanceau and Bouard [28], the solid line the numerical results of Rosenfeld et al. [15], and the dashed line those of Collins and Dennis [29]. Good agreement is obtained, especially at the initial stages of the flow evolution ($t < 8$). At time $t = 8$, our results agree well with the experimental data and that of Rosenfeld et al. At $t = 10$, all three numerical values are below the experimental value but our one is the closest. At $t = 12$, Collins and Dennis's value agrees the best with the experimental data, ours is slightly smaller and that of Rosenfeld et al. is yet smaller. Finally, the evolution of the separation length is computed using three different meshes and shown in Fig. 9, where the convergence of the numerical method is clearly demonstrated.

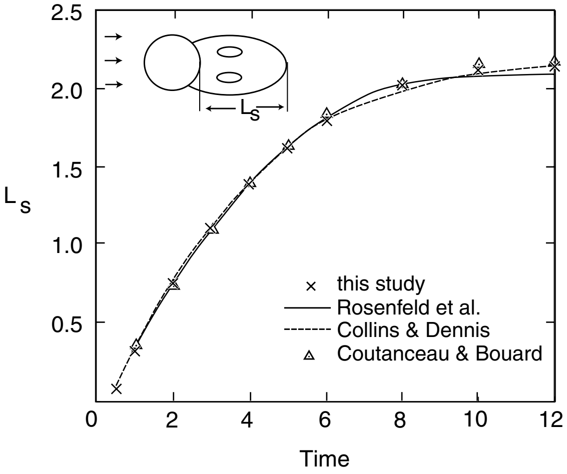


Fig. 8. Time evolution of the separation length behind the circular cylinder at *Re* = 40.

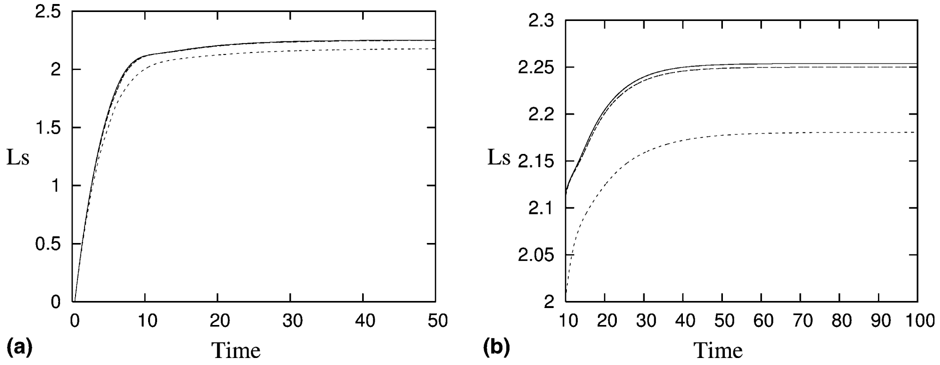


Fig. 9. (a) Time evolution of the separation length behind the circular cylinder at *Re* = 40. The solid line is computed on a 96 × 160 mesh, the dashed line on a 48 × 80 mesh and the dotted line on a 24 × 40 mesh. (b) A zoom between time 10 and 100.

4.2. Bubble dynamics

Bubble dynamics includes several well-studied free-surface problems. We first validate the spatial accuracy of our method against the well-known rising bubble problem. In this work, steady solutions are obtained in the framework of the rising bubble by setting the velocity at the far field equal to $-U_s$, where U_s is the final speed we prescribe. We adjust the gravity g in order to reach a steady state where the bubble stays at a same position. Fig. 10 summarizes our numerical simulations, where we have used a 64×64 mesh in all calculations, and the radius of the initial spherical bubble was unity while the outer radius of the computational domain was 50. This figure shows the bubble shapes for several Reynolds and Weber numbers and are close to identical with the numerical results of Ryskin and Leal [11]. The slight asymmetry seen in some of the dashed bubble shapes is an artifact, because the bubble shapes from Ryskin and Leal were traced from a scanned image.

The bubble rising from rest is computed in the framework at rest using the same gravity g as obtained in the above simulations. This is a time-dependent flow problem: while the bubble reaches a steady rising speed U_s at the end, its position and accordingly the numerical mesh change constantly. Fig. 11 shows superpositions of bubble shapes for $We = 2$ and $We = 4$ respectively at $Re = 10$. The solid lines represent the bubble shape in the framework of the rising bubble and the dashed lines in the framework at rest. There is no discernible difference. The difference in the final rising speed between the two sets of simulations is negligible (less than 0.0098%). This comparison provides a validation of the temporal accuracy of our method. Mass conservation is an important criteria characterising the quality of a numerical method for moving interface problems. Our method performs very satisfactorily in this matter. For instance, the maximum loss of mass in the two simulations of the bubble rising problem in the framework at rest is less than 0.12%, and 0.026% in the framework of rising bubble.

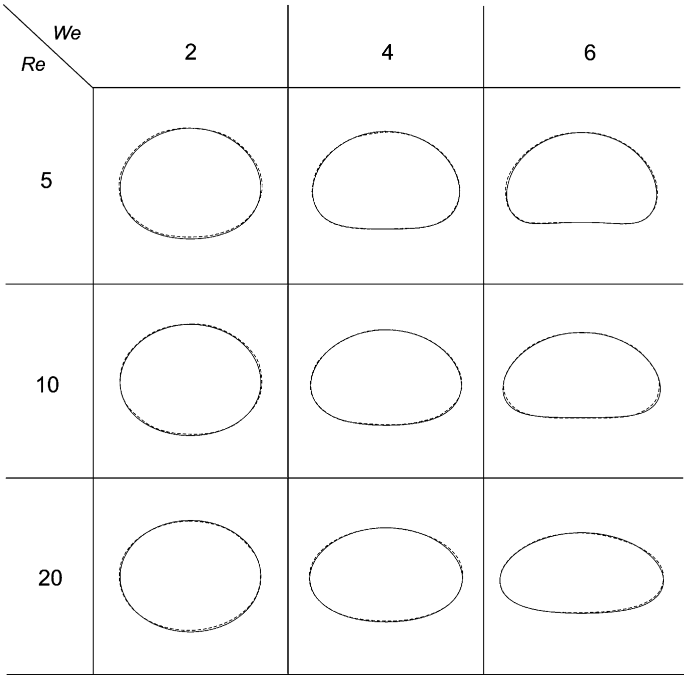


Fig. 10. Comparison between our numerical results (solid lines) and the numerical results of Ryskin and Leal [11] (dashed lines).

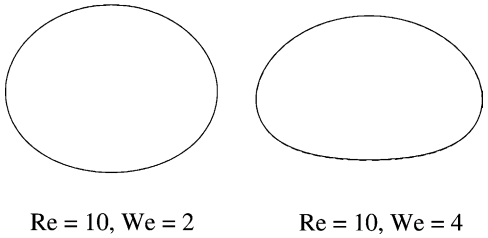


Fig. 11. Comparisons bubble shapes between simulations in the framework of the rising bubble (steady solutions) and the framework at rest (unsteady solutions). The bubble shapes of the unsteady solutions are shifted to best fit those of the steady solutions. The solid lines represent the bubble shape in the framework of the rising bubble and the dashed lines in the framework at rest. We cannot discern the dashed lines as the agreement is excellent.

The spatial and temporal accuracy of our method is finally validated against theoretical solutions of an unsteady oscillatory bubble initially at rest. In the classic normal-mode analysis [30], the interface of an oscillating bubble is described as

$$R(\theta,t) = R_0 + \epsilon_n P_n(\cos\theta)\sin(\omega_n t),$$

where R_0 is the unperturbed radius, P_n the Legendre polynomial of degree n, and

$$\omega_n^2 = (n+1)(n-1)(n+2)\frac{\gamma}{\rho R_0^3}$$

the frequency of oscillation. If the viscosity is taken into account, the amplitude decays as $\mathrm{e}^{-b_{nt}}$, where the rate of damping

$$b_n = (n+2)(2n+1)\frac{\nu}{R_0^2}.$$

This normal-mode analysis is, however, not suitable for the initial-value problem studied here [31]. The initial-value problem is best described by a solution obtained by Prosperetti [26] expressed by Laplace transform:

$$\tilde{a}(p) = \frac{1}{p}\left(a_0 + \frac{p\dot{a}_0 - \omega_n^2 a_0}{p^2 + 2b_0 p + \omega_n^2 + 2\beta b_0 p\tilde{Q}(p)}\right) \tag{24}$$

where $\tilde{a}$ is the Laplace transform of the amplitude, a_0 the initial amplitude, $\dot{a}_0$ its derivative, β defined by

$$\beta = \frac{n(n+2)}{2n+1} \tag{25}$$

and $\tilde{Q}$

$$\tilde{Q}(p) = -\frac{1}{1+\frac{1}{2}\Gamma(R_0\sqrt{p/\nu})}. \tag{26}$$

The geometrical function Γ is expressed in term of modified Bessel functions K of the second kind

$$\Gamma = q\frac{K_{n+1/2}(q)}{K_{n-1/2}(q)}.$$

Fig. 12(a) shows the temporal evolution of the amplitude of the second mode of deformation a bubble set in a liquid initially at rest. The physical parameters are $R_0 = 1$, $\rho = 1$, $\gamma = 1$, $\mu = 0.01414$ and $\epsilon_2 = 0.01$. Comparing our numerical results with the two theories above we find an excellent agreement with Prosperetti's

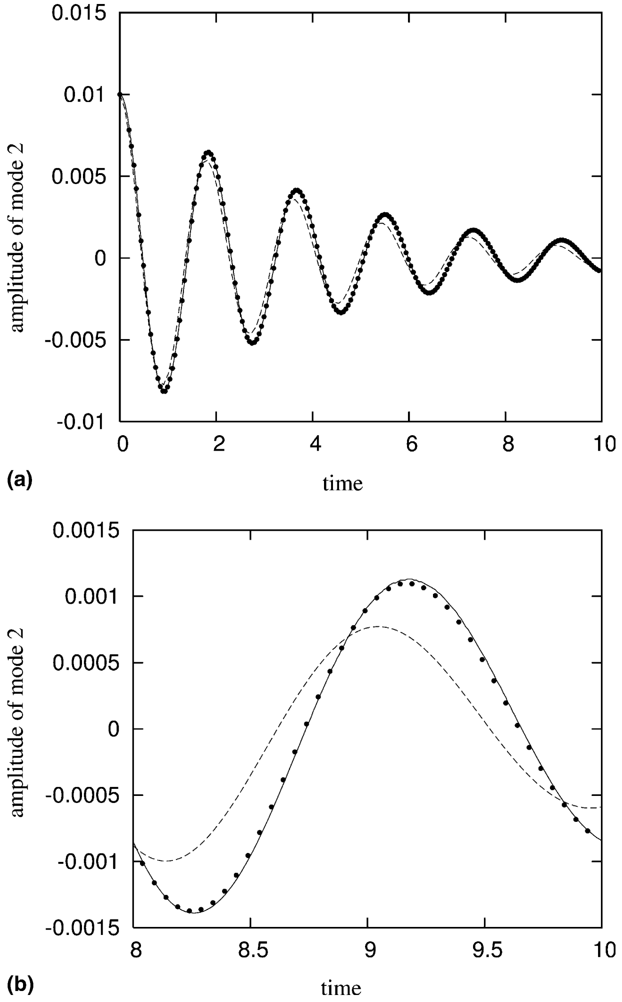


Fig. 12. Temporal evolution of the second mode of deformation of a bubble set in a liquid initially at rest. Our simulations use a mesh of size 128 × 128 (solid lines). The theoretical curves are obtained from a normal-mode analysis (the dashed lines) and from the exact solution to the initial-value problem (the black circles). (a) The second mode, and (b) a zoom on the second mode illustrating the difference between the normal-mode and initial-value solutions.

theory. Fig. 12(b) shows a close up at the sixth crest illustrating the difference between the normal-mode and initial-value solutions. We note that the difference between the approximate normal-mode solution and Prosperetti's solution is significant. On the other hand, the relative difference at this crest is about 3% between our simulation and Prosperetti's theory. The agreement on the evolution of mode 3 and mode 4 between our simulation and Prosperetti's theory is equally excellent (not shown). This test further confirms that our numerical method is accurate both in space and in time.

5. Interaction between a free surface and a solid structure

The motivation of this work is to develop a numerical method that we can use to study the surface locomotion of semi-aquatic insects. The characteristic weight of water striders is $W = 10^{-4}$ N. Walking and jumping on water poses no problem for water striders. In this work, we validate our numerical method on a simplified two-dimensional mechanical model consisting of a massless plate (foot) and spring (leg) supporting a point mass (body). A sketch of this model is shown in Fig. 13. We model the leg as an elastic spring with an elastic modulus k, connected to a flat plate of width $2L$ which initially rests on the water. In our model, the weight W of the strider is concentrated in the solid body above the spring. Following [32, p. 243], the static depth h of the plate compared to the unperturbed water level as well as the water surface are deduced theoretically for a periodic case.

This theoretical solution is expressed in terms of an elliptic integral which can only be computed numerically. Nevertheless, an analytical solution can be expressed in terms of elementary functions for the special case where the distance d between two plates is $+\infty$,

$$W = \frac{\gamma h}{l_c}\left[\left(1-\left(\frac{h}{2l_c}\right)^2\right)^{1/2} + \frac{2L}{l_c}\right]. \qquad (27)$$

The capillary length of water is $l_c = (\gamma/\rho g)^{1/2} \approx 2.72\times 10^{-3}$ m, where ρ is the density of water, and g is the gravity acceleration.

To simplify the problem, we study a symmetrical case where we assume the solid plate (foot) has only a vertical velocity, equal to that of the fluid below the plate:

$$v = V_{\text{plate}}. \qquad (28)$$

In a conventional problem, the boundary condition on the solid structure would be a Dirichlet condition imposed on the velocity. However, in the above system, we know the force acting on the plate instead of the velocity. The boundary condition on the solid plate is a balance between the force of the spring F_{spring} and the force F_{water} exerted on the plate by the water:

$$F_{\text{spring}} = F_{\text{water}}, \qquad (29)$$

and

$$F_{\text{water}} = \int_S \left(-p + 2\mu\frac{\partial v}{\partial y}\right)\mathrm{d}S + 2\gamma\cos(\phi), \qquad (30)$$

where S is the area of the plate, and ϕ the angle between the water surface at the contact point and the horizontal. A single step of our flow solver computes the velocity field using a given velocity V_{plate} on the solid plate. This velocity is, however, not known à priori. We employ a secant iterative method [33] to find V_{plate} such that Eq. (29) is satisfied. The motion of the solid body is governed by Newton's second law:

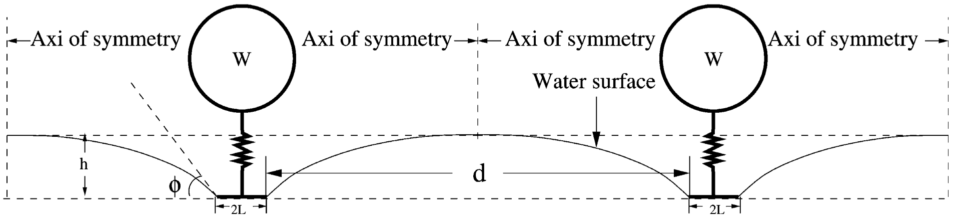


Fig. 13. A periodic array of mechanical water striders resting on a water surface. The mechanical water striders are drawn in thick solid lines, while the water surface in thine solid lines. d is the distance between the plates, and ϕ the angle between the water surface and the horizontal level at the contact point.

$$M\frac{\mathrm{d}V}{\mathrm{d}t} = F_{\text{spring}} - Mg = k\Delta l - Mg, \tag{31}$$

where M is the mass of the solid body, V its speed, and Δl is the compression of the spring. This ordinary equation is coupled with the flow solver. It is solved implicitly in order to obtain a stable numerical scheme.

5.1. Static solution

We first check the accuracy of our method against a static solution. To avoid numerical boundary conditions at the far field, we construct a periodic solution (see Fig. 13), using the theoretical results in [32, p. 243]. The equilibrium condition on the water surface in a gravitational field is

$$\frac{2z}{a^2} - \frac{z''}{(1+z')^{\frac{3}{2}}} = \text{constant}, \tag{32}$$

where $a = \sqrt{2\gamma/\rho g}$ is the capillary constant. Here we note z as the vertical coordinate (to be consistent with [32]). We can choose $z(0) = a^2 z''(0)/2$ at a symmetric point $x = 0$ where z' is zero. For this choice, the constant in the above equation is zero. Multiply Eq. (32) by z', and integrate it, we obtain

$$\frac{1}{(1+z')^{\frac{1}{2}}} = A - \frac{z^2}{a^2},$$

where we have $A = 1 + \frac{z(0)^2}{a^2}$. So the second derivative at the symmetric point is

$$z''(0) = \frac{2}{a}(A-1)^{\frac{1}{2}}. \tag{33}$$

Finally, the distance d between the two plates is expressed in term of an elliptic integral,

$$d = \frac{a}{2}\int_0^{\phi} \frac{\cos\xi \,\mathrm{d}\xi}{\sqrt{A - \cos\xi}}, \tag{34}$$

where ϕ the angle between the water surface and the horizontal level at the contact point.

It is easy to construct a static solution from the parameters A and ϕ. For $A = 1.002$ and $\phi = 0.5738$, the distance between two plates d is 8.0 (Eq. (34)), and from Eq. (33) the curvature at the symmetrical point $\kappa = 0.04472$. A static solution is constructed from the above parameter for $L = 1.0$, where $2L$ is the width of a plate. Using the symmetry of the solution, the computation is done in one half of the domain, for which a mesh of size 16×16 is shown in Fig. 14(a). A cubic spline is used to obtain a continuous smooth parametrization of the surface and to compute the surface curvature accurately. A third order forward finite-difference scheme, in consistent with the accuracy of cubic spline, is used at the contact points to estimate the first derivatives of the parametrization. The numerical error of the curvature at the symmetric point is shown in Fig. 14(b) in a logarithm scale for several mesh size and a second order convergence of the cubic spline is achieved.

An interesting test is the verification of the static solution constructed above. Two questions follow, the stability of the approximate solution and its numerical convergence to the theoretical solution. A numerical test is done for $\gamma = 1.0$, $\rho = 1.0$, $g = 0.5$ and $\mu = 0.2$, with an equilibrium weight $W = 0.9454$ sitting on the solid plate (corresponding to the stiff spring case of $k = +\infty$), where

$$W = \gamma\sin(\phi) + L[\rho g h - \kappa(0)],$$

and the water level $h = z(0) - z(\frac{d}{2})$ is 0.715894.

We start the simulation with the theoretical solution at the rest. Because of numerical errors on a finite mesh, the numerical solutions deviate from this solution, and oscillate around new steady solutions

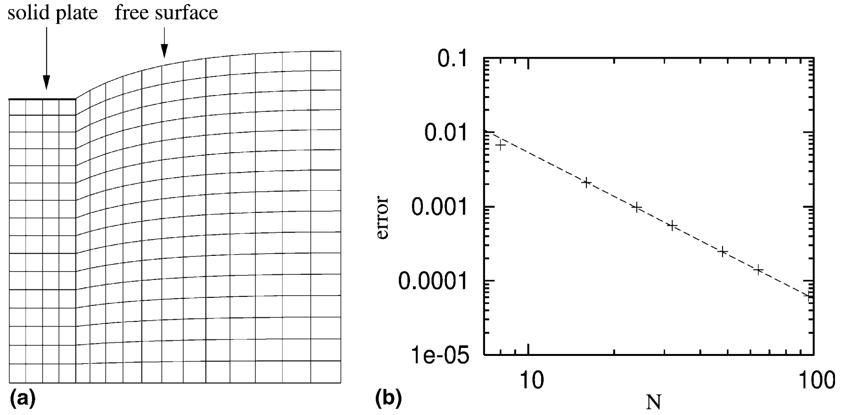


Fig. 14. (a) A mesh of size 16 × 16. (b) Second order convergence of the curvature at the symmetric point. The '+' symbol represents the numerical error as function of the number of grid points $N \times N$ in a logarithm scale, while the dashed line a power function with an exponent −1.97.

depending on the numerical meshes. The time evolution of the solid plate position is shown in Fig. 15(a) for four meshes, and they all reach a steady position at a large time. The solid (horizontal) line is the theoretical value. The numerical error as function of the number of grid points $N \times N$ is shown in Fig. 15(b) in a logarithm scale and a second order convergence is demonstrated.

5.2. Dynamical solution

The ability of our numerical method to handle the interaction between a solid structure and a free-surface is further demonstrated on a dynamical problem. In this subsection, we are interested in a jumping process of a single strider jumping on the water surface. For a strider of weight of order 10^{-4} N, the depth h is small compared to the capillary length l_c, typically $h/l_c = 0.05$. For our application, L is of order of the leg diameter, ≈8×10^{-5} m, and so typically $L/l_c = 0.015$. In the equilibrium state, Eq. (27) is therefore approximated by

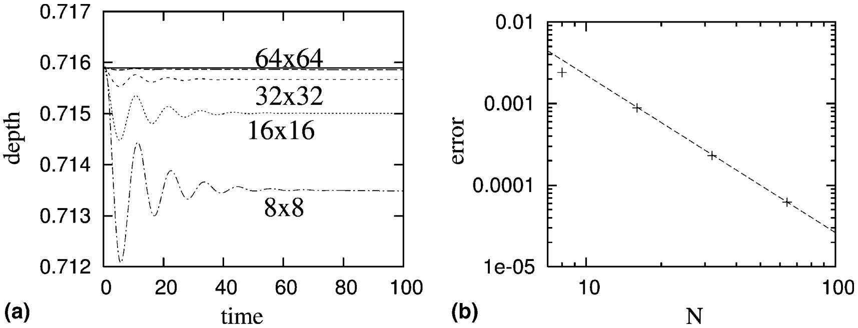


Fig. 15. (a) Evolution of the solid plate position for different mesh sizes comparing to the theoretical value of the static solution. The solid (horizontal) line is the theoretical value. (b) Second order convergence of the solid plate position at the final time. The '+' symbol represents the numerical error as function of the number of grid points $N \times N$ in a logarithm scale, while the dashed line a power function with an exponent −1.93.

$$F_s \approx \left(\frac{\gamma}{l_c}\right)h = (\rho g \gamma)^{1/2}h. \tag{35}$$

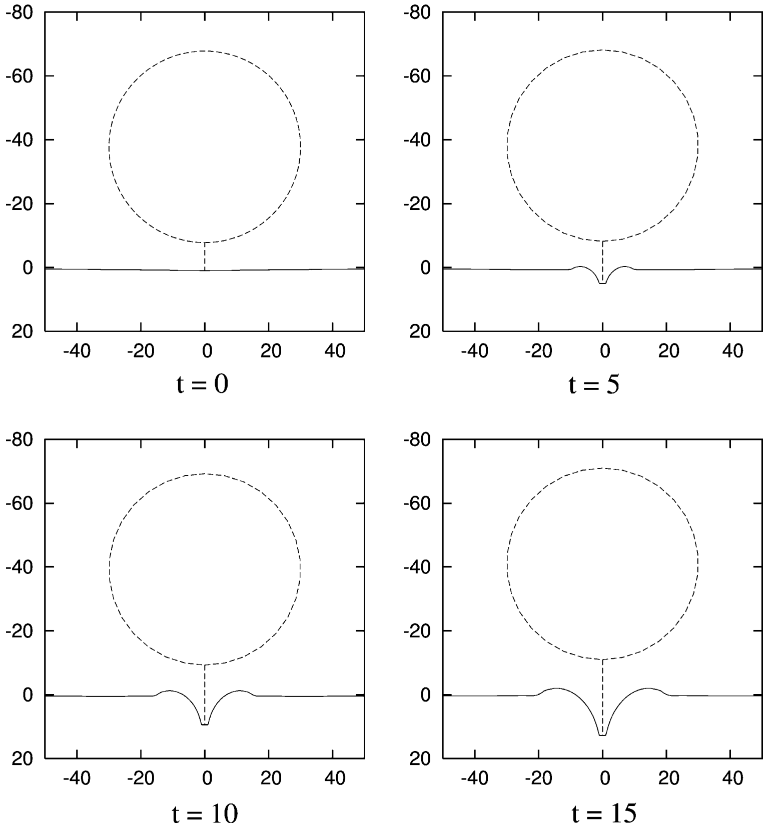
The quantity γ/l_c has the same units as the elastic modulus of the spring, and it provides a way to measure the strength of the spring relative to the surface tension:

$$\xi = \frac{kl_c}{\gamma}. \tag{36}$$

To support the weight, the spring is compressed to an equilibrium displacement $e = W/k$. If an additional compression $c < e$ is applied to the spring of the model strider when placed on a solid surface, then the weight will oscillate around its equilibrium position. If $c > e$, the strider will separate from the surface. We define the ratio of the additional compression c to the equilibrium compression e as

$$\eta = \frac{c}{e}.$$

In this subsection, we consider a similar problem when such a mechanical strider is posed on a water surface. We take the density of water $\rho = 1.007 \times 10^3$ kg/m^3 and the surface tension to be $\gamma = 7.28 \times 10^{-2}$ N/m. Using the radius of the strider leg ($\approx 4 \times 10^{-5}$ m) as a length scale, we find that the capillary time is given by $T = (\rho L^3/\gamma)^{1/2} = 2.957 \times 10^{-5}$ s, the ratio of gravitational to capillary forces is $gT^2/L = 2.12 \times 10^{-4}$, and the



t = 0

t = 5

t = 10

t = 15

Fig. 16. The motion of a mass-spring system which is initially under compression and resting on water. The physical parameters for this computation are $L = 1$, $h = 1$, $c/e = 100$, and $kl_c/\gamma = 2$.

ratio of viscous to capillary stress is $\mu/\sqrt{\rho\gamma L} = 0.0185$, where the dynamic viscosity of water is taken to have value $\mu = 1.002 \times 10^{-3}$ kg/m s.

We compute the motion of a mass-spring system which is initially under compression and resting on water. The physical parameters for this computation are $L = 1$, $h = 1$, $c/e = 100$, and $kl_c/\gamma = 2$. The simulation is performed in a box of dimensionless size 400 × 400 so that several capillary lengths are included and a symmetry condition at the far field is a good approximation. At the initial time, the water surface is almost flat (Fig. 16). As the initial compression $c/e = 100$ is very large, the interaction process is in a non-equilibrium regime. In the successive pictures ($t = 5$, 10 and 15), Fig. 16 illustrates a formation of a crater where the solid plate is pushed down, and that a local disturbance propagates outwards from the plate, with negligible disturbance in the far field. The velocity field in the fluid associated with this disturbance is shown in Fig. 17 at time $t = 1$, 2, 3 and 4, showing the water is pushed down below the solid plate and returns up beside to form a crater. There is no theoretical solution for this flow problem. The comparison of our computations on meshes of different sizes is shown in Fig. 18. A clear convergence is observed as the mesh is refined. The discrepancy occurs on the outgoing wave front, which is due to the poor resolution of the coarse meshes. Nevertheless, on the crater part where the solid plate is pushed down, the computation on the coarsest mesh agrees very well with that of the finest mesh. The dynamics on this later region is

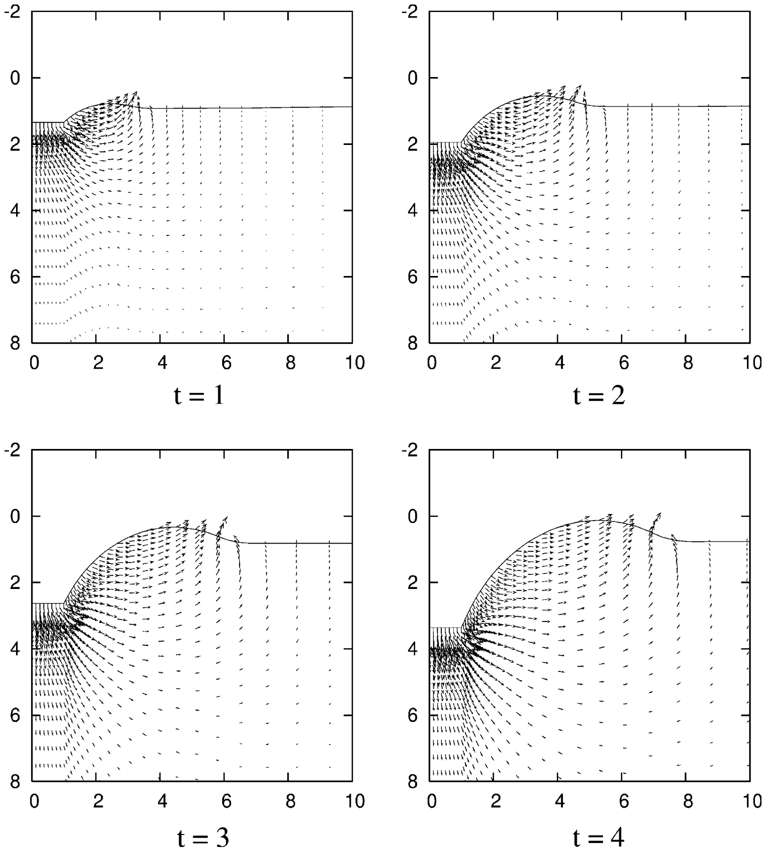


Fig. 17. The velocity field at time $t = 1$, 2, 3 and 4. The physical parameters for this computation are $L = 1$, $h = 1$, $c/e = 100$, and $k\ l_c/\gamma = 2$.

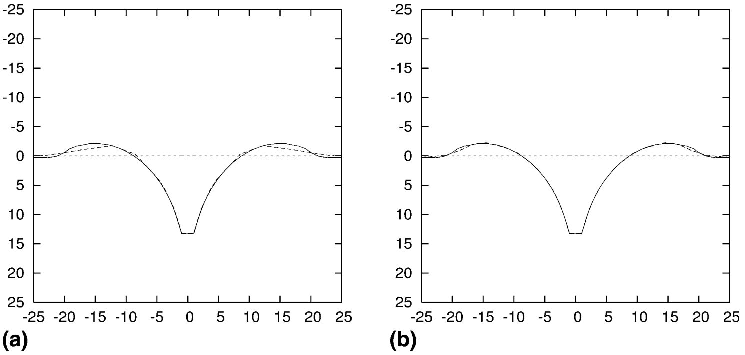


Fig. 18. Positions of the free surface and the solid plate at time $t = 15$. The solid line represents the computation on a high resolution mesh 128×128, and the dashed line (a) on a coarse mesh 32×32, and (b) on an intermediate mesh 64×64. The dotted line represent the initial position. The physical parameters in this computation are $L = 1$, $h = 1$, $c/e = 100$, and $kl_c/\gamma = 2$.

the utmost factor determining the response of the mechanical strider. The accuracy of our method is therefore good enough to capture the main feature of this problem even on a coarse mesh.

6. Minimum compression for jumping off water

Our numerical method is used to explore the critical compression required for a model mechanical strider to jump off the water surface. When the mechanical strider is placed on a solid surface, it will take off if the spring becomes completely relaxed (the compression falls to zero). This occurs if the extra compression exceeds the equilibrium value, $\xi > 1$. When the mechanical strider attempts to jump off water, the water beneath the foot also moves and absorbs part of the energy. The minimum extra compression required to jump off the surface therefore increases. The separation from the water surface depends on how the leg is detached from the water. Since the body and legs of the water strider are covered by thousands of hairs, its legs are effectively non-wetting. As a simplification, we therefore approximate the detailed detachment process by applying the same criterion that the spring compression falls to zero, as for the solid surface. We have performed a series of computations in order to delineate the minimum compression for separation from the surface. Fig. 19 shows the variation of the minimum compression for jumping as a

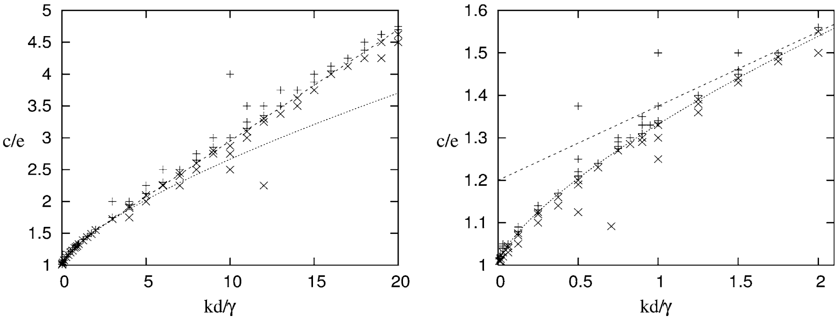


Fig. 19. Diagram of ejection based on the spring model. The equilibrium height is $h = 1$ and the plate width $L = 1$. The '+' symbol represents ejection and the '×' symbol non-ejection. The two regimes are separated by a straight line $0.175(kl_c/\gamma) + 1.2$ (the dashed line) for stiff springs (top figure), and by a power law $0.332\ (kl_c/\gamma)^{0.7} + 1.0$ with an exponent 0.7 (the dotted line) for weak springs (bottom figure). e is the equilibrium compression of spring, and c the extra compression, k the spring constant, $l_c = (\gamma/\rho g)^{1/2}$ the capillary length, and γ the surface tension constant.

function of the dimensionless spring modulus for a strider of weight equivalent to a depth $h = 1.0$ and a width $L = 1$. The results are surprisingly simple. The minimum compression for jumping varies linearly with spring constant (the dashed line) for stiff springs, and follows a power law with an exponent 0.7 (the dotted line) for weak springs. The straight line is

$$0.175(kl_c/\gamma) + 1.2,$$

and the power law is

$$0.332(kl_c/\gamma)^{0.7} + 1.0.$$

The quantity kl_c/γ is a measure of the strength of the spring compared to the surface tension of the water surface. When it is equal to zero, the spring is extremely weak and the minimum compression converges to 1 as the spring becomes weaker and weaker relative to the surface tension, corresponding to the case of an effectively solid surface.

It is interesting to examine the evolution of different forces versus time. Two cases are shown in Fig. 20, the left figure for a weak spring ($kl_c/\gamma = 0.125$ and $c/e = 1.08$) and the right figure for a strong spring ($kl_c/\gamma = 15$ and $c/e = 3.875$). Both calculations correspond to cases in which the mechanical strider is just able to separate from the water surface. In these figures, time is rescaled in terms of the surface tension relaxation time $(\rho L^3/\gamma)^{1/2}$. We observe that although the spring force (represented by the solid line) is initially large, it is quickly balanced by the capillary force through deformation of the interface close to the plate (represented by the dashed line), and they come into balance after a dimensionless time of about 2 for both cases. The equilibrium capillary force computed by Eq. (35) is represented as the dotted line, which is very different form the real surface tension force. This shows that the whole jumping process is in a non-equilibrium regime.

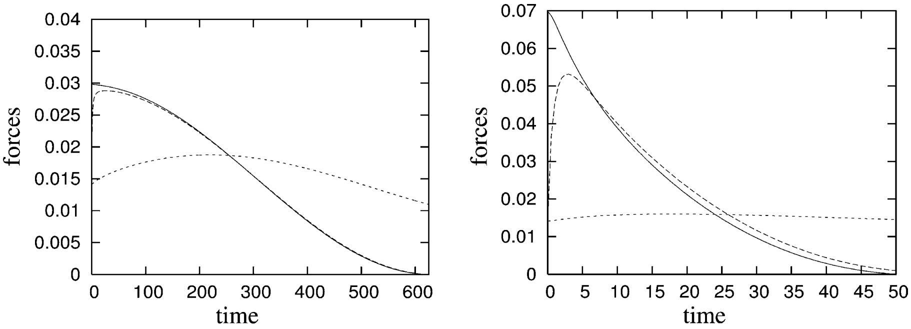


Fig. 20. The evolution of forces against time. The spring is weak on the left figure ($kl_c/\gamma = 0.125$ and $c/e = 1.08$) while it is strong on the right one ($kl_c/\gamma = 15$ and $c/e = 3.875$). The solid line represents the elastic force of spring, the dashed line the capillary force and the dotted line the equilibrium force computed by Eq. (35). This figure shows that the jumping process is in a non-equilibrium regime.

Table 1
The slope a of the linear straight and the coefficient b of the power law for different weights

h	0.5	1.0	2.0	3.0	4.0	5.0	6.0
a	0.116	0.175	0.260	0.328	0.39	0.45	0.495
b	0.255	0.332	0.44	0.52	0.585	0.63	0.69
β	0.255	0.25494	0.2560	0.257	0.256	0.250	0.2558
α	1.0	0.663	0.446	0.354	0.297	0.258	0.234

The two regimes of ejection and non-ejection are separated by a straight line $a(kl_c/\gamma) + 1.2$ for stiff springs, and by a power law $b(kl_c/\gamma)^{0.7} + 1.0$ for weak springs. The rescaled coefficient β is nearly a constant.

6.1. Effect of the depth

We have computed the critical values for different weights that are equivalent to equilibrium depths h ranging from 0.5 to 6. The key result of this work is that the minimum compression for the strider to separate from the surface varies linearly with spring constant $a(kl_c/\gamma) + 1.2$ for stiff springs, and by a power law $b(kl_c/\gamma)^{0.7} + 1.0$ for weak springs. The linear slope a and the coefficient b of the power law are shown in Table 1. These curves collapse into a master curve if we use a proper scaling: using the curve for $h = 0.5$ as a reference curve for example, we change the variable for each curve to $\alpha kl_c/\gamma$, where α is a constant corresponding to the curve such that the rescaled straight line coincides with the reference one. α is shown in the last row of Table 1. Then the rescaled coefficient for the power law will be $\beta = b\alpha^{0.7}$. As shown in Table 1, β is nearly a constant 0.255. A graphic representation is Fig. 21(a). The dependency of α as function of h is a power function with an exponent roughly equals to −0.6. This is shown in Fig. 21(b). This suggests that a proper scaling also involves the equilibrium depth h, and that a universal curve for the minimum compression depends on $kl_c^{1.6}/\gamma h^{0.6}$.

6.2. Effect of the plate width

We have also investigated the effect of the width of the plate (foot) for a fixed high $h = 1$, and our observations that a linear law $a(kl_c/\gamma) + f$ for strong springs and a power law $b(kl_c/\gamma)^{0.7} + 1.0$ for weak springs applies for small values of L ($L < 2$). The coefficients are listed in Table 2: which suggests convergence as $L \to 0$. As the plate width increases beyond values of order 1, an increasing amount of energy is absorbed by the fluid before the strider can take off the water surface leading to a change of the functional form of the scaling laws for the minimum compression.

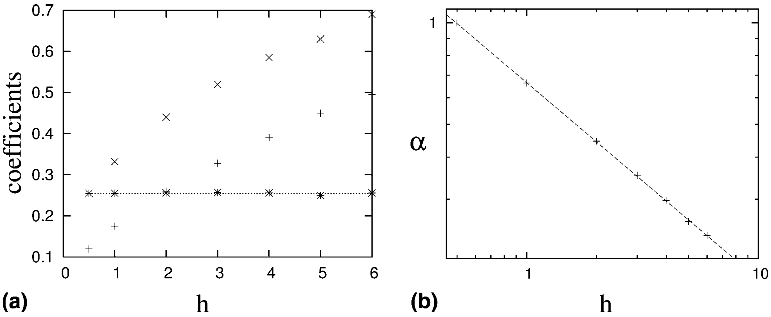


Fig. 21. (a) The slope a (represented by the '+' symbol) of the linear straight and the coefficient b (represented by the '×' symbol) of the power law as function of the depth h. The star symbol represents the rescaled coefficient β of the power law. The value of the dotted-line is 0.255. (b) α (represented by the '+' symbol) as function of the depth h is fitted by a power function $0.665h^{-0.583}$. Both axes are represented in the log scale.

Table 2
The slope a of the linear straight and the coefficient b of the power law for different widths

L	0.1	0.2	0.5	1.0	1.5	2.0
a	0.193	0.192	0.185	0.175	0.158	0.14
b	0.345	0.34	0.335	0.332	0.33	0.32
f	1.2	1.2	1.2	1.2	1.25	1.3

7. Conclusion

An ALE moving mesh method for solving two-dimensional and axisymmetric moving-boundary problems, including the interaction between a free-surface and a solid structure, is presented. This method employs a body-fitted grid system where the gas–liquid interface and solid–liquid interface are lines of the grid system. The main novelties of this work lie in the treatment of boundary conditions. For reasons of simplicity and clarity, we employ an explicit projection method, and all boundary conditions are to be satisfied on time level n. Based on a Finite-Volume formulation, complicated dynamic boundary conditions are incorporated naturally and accurately on the free-surface. On the solid structure, a usual boundary condition is a Dirichlet condition imposed on the velocity. However, the interaction between a liquid and a solid often requires boundary conditions to be imposed on the forces acting on the solid structure. In this work, a single step of our flow solver uses a given velocity on the solid structure. This velocity is, however, not known à priori. We therefore employ an iterative method to find the correct velocity on the solid structure such that the force balance is satisfied there. The pressure equation needs special care at the transition point where the free-surface and the liquid–solid interface meet. This is nicely treated by converting the discrete pressure equation into a unified form (Eq. (19)), which is solved by an efficient multigrid method. This work is part of our effort to incorporate complicated physics involving moving boundaries in a general and flexible framework. We have included new abilities, such as handling a viscoelastic fluid and the Marangoni effect induced from a surfactant, which we will present in a separate paper.

Our method is validated on the uniform flow passing over a cylinder (a two-dimensional flow with a solid structure) and several problems of bubble dynamics (axi-symmetrical flows with a free surface) for both steady and unsteady flows. Good agreement with other theoretical, numerical and experimental results is obtained. An incentive of this work is to develop a numerical method which can be used to study the hydrodynamics underlying the surface locomotion of semi-aquatic insects. Further validation is therefore the investigation of a two-dimensional mechanical strider interacting with a water surface. We have constructed a theoretical static solution on which we studied the convergence of our method. Further demonstration of the abilities of the method is shown on a dynamical problem of jumping on a water surface. The main result of this investigation is the computation of the minimum spring compression required to separate from the water surface starting from an initial state of rest. The study is motivated by the bio-locomotion problem of the insects which jump on water, although our spring-mass model is a much simplified analogue designed to capture some of the fundamental physical constraints on the motion. We find the critical compression required to jump from the water varies linearly with spring constant for stiff springs and algebraically with exponent 0.7 for weak springs. We are currently working on a simplified dynamical model for this process.

Acknowledgment

JL's research is partially supported by EPSRC grant GR/S16348/01.

References

[1] D.L. Hu, B. Chan, J.W.M. Bush, The hydrodynamics of water strider locomotion, Nature 424 (6949) (2003) 663–666.

[2] G. Tryggvason, S.O. Unverdi, Computations of three-dimensional Rayleigh–Taylor instability, Phys. Fluids A 2 (1990) 656–659.

[3] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, J. Comput. Phys. 100 (1992) 25–37.

[4] C.W. Hirt, B.D. Nichols, Volume of fluid VOF method for the dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201–225.

[5] J. Li, Calcul d'Interface Affine par Morceaux (Piecewise linear interface calculation), C. R. Acad. Sci. Paris 320 (IIb) (1995) 391–396.

[6] D. Gueyffier, J. Li, A. Nadim, R. Scardovell, S. Zaleski, Volume-of-fluid interface tracking and smoothed surface stress methods applied to multiphase flow and pendant drop pinching, J. Comput. Phys. 152 (1999) 423–456.

[7] R. Scardovelli, S. Zaleski, Direct numerical simulation of free surface and interfacial flow, Ann. Rev. Fluid Mech. 31 (1999) 567–604.

[8] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.

[9] S. Popinet, S. Zaleski, A front-tracking algorithm for the accurate representation of surface tension, Int. J. Numer. Methods Fluids 30 (1999) 775–793.

[10] Y. Renardy, M. Renardy, Prost: a parabolic reconstruction of surface tension for the volume-of-fluid method, J. Comput. Phys. 183 (2002) 400–421.

[11] G. Ryskin, L.G. Leal, 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. 148 (1984) 19–36.

[12] B. Maury, Direct simulations of 2d fluid-particle flows in biperiodic domains, J. Comput. Phys. 156 (2) (1999) 325–351.

[13] P. Fast, M.J. Shelley, A moving overset grid method for interface dynamics applied to non-Newtonian hele-shaw flow, J. Comput. Phys. 195 (1) (2004) 117–142.

[14] A.J. Chorin, A numerical method for solving incompressible viscous flow problems, J. Comput. Phys. 2 (1967) 12–26.

[15] M. Rosenfeld, D. Kwak, M. Vinokur, A fractional step solution method for the unsteady incompressible Navier–Stokes equations in generalized coordinate systems, J. Comput. Phys 94 (1991) 102–137.

[16] F.H. Harlow, J.E. Welsh, Numerical calculation of time-dependent viscous incompressible flow with free surface, Phys. Fluids 8 (1965) 2182–2189.

[17] R. Peyret, T.D. Taylor, Computational Methods for Fluid Flow, Springer-Verlag, 1990.

[18] C.M. Rhie, W.L. Chow, A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation, AIAA-83-0260, 1983.

[19] Joel H. Ferziger, M. (Milovan) Peric, Computational Methods for Fluid Dynamics, Springer, 1966.

[20] Y. Zang, R.L. Street, J.R. Koseff, A non-staggered grid fractional step method for time-dependent incompressible Navier–Stokes equations in curvilinear coordinates, J. Comput. Phys. 114 (1994) 18–33.

[21] J. Li, Numerical resolution of Navier–Stokes equation with reconnection of interfaces. Volume tracking and application to atomization, PhD thesis, University of Paris VI, 1996.

[22] R.L. Sani, P.M. GreshoIncompressible Flow and the Finite Element Method, vol. 2, John Wiley, 2000.

[23] P.J. Zwart, G.D. Raithby, M.J. Raw, The integrated space–time finite volume method and its application to moving boundary problems, J. Comput. Phys. 154 (2) (1999) 497–519.

[24] R.S. Bernard, H. Kapitza, How to discretize the pressure gradient for curvilinear MAC grids, J. Comput. Phys. 1992 (1992) 288–298.

[25] J.F. Thompson, Z.U.A. Warsi, C.W. Mastin, Numerical Grid Generation: Foundations and Applications, North-Holland, 1985.

[26] A. Prosperetti, Free oscillations of drops and bubbles: the initial-value problem, J. Fluid Mech. 100 (1980) 333–347.

[27] L. Fuchs, N. Tillmark, Numerical and experimental study of driven flow in a polar cavity, Int. J. Numer. Methods Fluids 5 (1985) 311–329.

[28] M. Coutanceau, R. Bouard, Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. Steady flow, J. Fluid Mech. 79 (1977) 231–256.

[29] W.M. Collins, S.C.R. Dennis, Flow past an impulsively started circular cylinder, J. Fluid Mech. 60 (1973) 105–127.

[30] H. Lamb, Hydrodynamics, Cambridge University Press, 1932.

[31] S. Popinet, S. Zaleski, Bubble collapse near a solid boundary: a numerical study of the influence of viscosity, J. Fluid Mech. 464 (2002) 137–163.

[32] L.D. Landau, E.M. Lifshits, Fluid Mechanics, Pergamon, Oxford, 1987.

[33] H. Press William, A. Teukolsky Saul, T. Vetterling William, P. Flannery Brian, Numerical Recipes in FORTRAN: The Art of Scientific Computing, Cambridge University Press, 1996.