

Numerical Simulation for Drop Impact on Textured Surfaces



Martina Baggio and Bernhard Weigand

Abstract Superhydrophobic surfaces with small-scale features have recently gained interest, because impacting droplets may bounce-off faster with respect to a flat superhydrophobic surface. For such surfaces the correct numerical prediction of the impact phenomena is very difficult. Our goal is the numerical study of drop impact on such surfaces using Free Surface 3D (*FS3D*), our in-house code for the simulation of incompressible multi-phase flows. Until recently, *FS3D* was not able to represent the interaction of a droplet with a complex textured solid surface. In this work, we show how we added this feature to the code by implementing the representation of embedded arbitrary-shaped boundaries using a Cartesian grid. Two approaches were developed; a preliminary simplified approach and an ultimate, more rigorous one. We discuss both implementations and we show a comparison of the two approaches for a test case. The results show that the predictions for impact dynamics of the two approaches slightly differ. Although, the simplified approach shows only small errors in mass conservation, it is fundamentally not conservative. With the introduction of a new approach we were able to improve the conservativeness of our simulations.

1 Introduction

Superhydrophobic surfaces are of great technical interest because of their non-wetting behavior. Recently, it has been shown that superhydrophobic surfaces with macro-scale features with dimensions between 10^{-1} and 10^0 mm can enhance water repellency by reducing the contact time between a water droplet and the solid surface. In particular, Bird et al. [1] have shown that the contact time for a droplet impacting on a superhydrophobic surface with a ridge was shorter than the contact time on a flat superhydrophobic surface because of the changed impact morphology. Gauthier et al. [2] found a relationship between contact time and number of liquid

M. Baggio (✉) · B. Weigand

Institute of Aerospace Thermodynamics, University of Stuttgart, Pfaffenwaldring 31, 70569 Stuttgart, DE, Germany

e-mail: martina.baggio@itlr.uni-stuttgart.de

© Springer Nature Switzerland AG 2020

G. Lamanna et al. (eds.), *Droplet Interactions and Spray Processes*,
Fluid Mechanics and Its Applications 121,

https://doi.org/10.1007/978-3-030-33338-6_10

sub-units formed during impact for droplet impacting on a superhydrophobic surface with a wire. The same trend between contact time and number of liquid sub-units was reproduced by Regulagadda et al. [3] for a drop impacting on the top and between superhydrophobic triangular ridges. The majority of literature dealing with contact time reduction on superhydrophobic surfaces with submillimetric- and millimetric-scale features consists of experimental studies. Chantelot et al. [4] performed both experiments and lattice Boltzmann simulations for drop impact on a superhydrophobic surface with a small spherical feature. Shen et al. [5] have presented experimental and numerical results for a drop bouncing on short cones. More numerical research exists for drop impact on superhydrophobic curved surfaces. Khojasteh et al. [6] have studied drop impact on hydrophobic and superhydrophobic spheres of different diameters. They used a Level-Set method for their simulations. Liu et al. [7] studied experimentally and numerically with a lattice Boltzmann method drop rebound on a cylindrical surface and analyzed the total momentum distribution in the directions parallel and perpendicular to the surface curvature. A numerical study on drop impact on hydrophobic and superhydrophobic cylinders at different wettabilities and impact velocities was also performed by Liu et al. [8]. It is apparent that only very few numerical research has dealt with drop impact on superhydrophobic surfaces with small scale features. This manuscript explain the method used in *Free Surface 3D (FS3D)* for predicting drop impact on textured surfaces. *FS3D* is a program for the Direct Numerical Simulation (DNS) of incompressible multi-phase flows, which was originally created at the Institute of Aerospace Thermodynamics at the University of Stuttgart and it is continuously being improved with new features. Up to now, *FS3D* has successfully been used for the prediction of various phenomena such as evaporation of oscillating droplets [9], drop collisions impact [10], jet break-up [11], ice formation [12] and sublimation [13]. Rauschenberger et al. [14] implemented a method to represent the motion of rigid particles immersed in a continuous fluid phase. However, until recently *FS3D* could not handle the motion of a fluid interface on a rigid body of arbitrary shape. Therefore, a prerequisite of our study was a method to represent arbitrary-shaped boundaries in a Cartesian grid. In this paper, after an illustration of *FS3D*'s numerical fundamentals, we will discuss two methods for the implementation of solid boundaries embedded in a Cartesian grid; a first approximate method and a second more rigorous method inspired by the work of Popinet [15]. With the first method, which was developed as a simplified approach, we introduced the necessary data structures and routines that later enabled us to develop the more rigorous method. After having illustrated the concepts on which the preliminary approach is based, we will discuss its problematics which led to the development of the second approach. After this, the second more rigorous method will be explained. Finally in Sect. 4 we compare results for a water drop impacting on a superhydrophobic surface with macro-ridges obtained with the two approaches.

2 Numerical Methods

In this section, the numerical fundamentals of *FS3D* are illustrated briefly; the work of Eisenschmidt et al. [16] provides a broader overview. Here we focus on the case of an incompressible Newtonian fluid of a single liquid phase immersed in a continuous gas phase. The flow is assumed to be isothermal without phase change. Thus, the energy equation does not need to be considered. The governing equations of the flow are then the conservation of mass:

$$\nabla \cdot \mathbf{u} = 0, \quad (1)$$

and momentum transport:

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot \mathbb{S} + \rho \mathbf{g} + \mathbf{f}_\sigma. \quad (2)$$

The \mathbf{f}_σ term in Eq. (2) represents the surface tension force per unit volume which is different from zero only at the interface. This term is modeled with the Continuum Surface Stress (CSS) model of Lafaurie et al. [17]. A further equation is necessary for interface tracking. In *FS3D*, this equation is obtained with the Volume of Fluid (VOF) method of Hirt and Nichols [18]. The liquid disperse phase is represented by a colour function $\chi(\mathbf{x})$ defined as follows:

$$\chi(\mathbf{x}) = \begin{cases} 1 & \text{inside the liquid phase} \\ 0 & \text{outside the liquid phase} \end{cases}. \quad (3)$$

It is assumed that each material property of the flow ψ , as for example density ρ or dynamic viscosity μ , is constant within each phase and is given by

$$\psi(\mathbf{x}) = \psi_f \chi(\mathbf{x}) + (1 - \chi(\mathbf{x})) \psi_g, \quad (4)$$

where ψ_f and ψ_g are the material property constant values in the liquid and gas phase, respectively. The equation for interface tracking is obtained by considering the volume fraction $f = (1/\Omega) \int_\Omega \chi(\mathbf{x}) d\mathbf{x}$, where Ω is an arbitrary control volume. The transport equation for f is:

$$\frac{\partial f}{\partial t} = -\nabla \cdot (f \mathbf{u}). \quad (5)$$

In a space discretized domain, f_i denotes the liquid volume fraction within the computational cell of index $\mathbf{i} = i\mathbf{e}_1 + j\mathbf{e}_2 + k\mathbf{e}_3$. The interface is then located by identifying those cells in which $0 < f_i < 1$ and reconstructed with the Piecewise Linear Interface Calculation (PLIC) scheme of Rider and Kothe [19]. Thus, in each interface cell

the interface is approximated by a plane of direction $-\hat{\mathbf{n}} = \nabla f / \|\nabla f\|$ and position determined analytically by the values of f .

2.1 FS3D's Advection Scheme

In *FS3D* two different methods are implemented for the numerical treatment of advection, a split and an unsplit scheme. Here the discussion is limited to the split scheme, since it was used for the presented results in this paper. Let us consider the advection equation for a generic scalar variable ϕ :

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0, \quad (6)$$

which can also be written as:

$$\frac{\partial \phi}{\partial t} = -\nabla \cdot (\phi \mathbf{u}) + \phi \nabla \cdot \mathbf{u}. \quad (7)$$

The variable ϕ can be the fluid volume fraction f or a component of momentum $\rho \mathbf{u} = \rho(u\mathbf{e}_1 + v\mathbf{e}_2 + w\mathbf{e}_3)$. The advection of ϕ is then carried out in each direction separately in three different steps (or sweeps) [19, 20]. Time integration of Eq. (7) is carried out subsequently for each sweep. For example, in \mathbf{e}_1 direction and for the first sweep one obtains:

$$\frac{\phi^* - \phi^n}{\Delta t} = -\frac{\partial(\phi u)}{\partial x} + [(1 - \beta)\phi^n + \beta\phi^*] \frac{\partial u}{\partial x}, \quad (8)$$

where the superscript * indicates an intermediate auxiliary time step. The second term on the right hand side is the divergence correction [19, 20], and $0 \leq \beta \leq 1$ a coefficient which indicates its implicit or explicit nature [20]. Space discretization of Eq. (7) leads to the discretized sweep equations. For example, in \mathbf{e}_1 -direction:

$$\Omega \frac{\bar{\phi}_{i-\frac{1}{2}}^* - \bar{\phi}_i^n}{\Delta t} = -A_{\mathbf{e}_1} \left\{ \left(F_{i+\frac{1}{2}\mathbf{e}_1}(\mathbf{u}, \phi) - F_{i-\frac{1}{2}\mathbf{e}_1}(\mathbf{u}, \phi) \right) + [(1 - \beta)\bar{\phi}_i^n + \beta\bar{\phi}_{i-\frac{1}{2}}^*] \left(u_{i+\frac{1}{2}\mathbf{e}_1} - u_{i-\frac{1}{2}\mathbf{e}_1} \right) \right\}, \quad (9)$$

where $\Omega = (\Delta x \mathbf{e}_1 \times \Delta y \mathbf{e}_2) \cdot \Delta z \mathbf{e}_3 = \Delta x \Delta y \Delta z$ is a Cartesian control volume, $A_{\mathbf{e}_1} = \|\Delta y \mathbf{e}_2 \times \Delta z \mathbf{e}_3\| = \Delta y \Delta z$ its face perpendicular to \mathbf{e}_1 . In Eq. (9), it was assumed that $\int_{\Omega} \phi \nabla \cdot \mathbf{u} dx \approx \bar{\phi} \int_{\Omega} \nabla \cdot \mathbf{u} dx = \bar{\phi} [\Delta y \Delta z (u_{i+\frac{1}{2}\mathbf{e}_1} - u_{i-\frac{1}{2}\mathbf{e}_1}) \mathbf{e}_1 + \Delta z \Delta x (v_{i+\frac{1}{2}\mathbf{e}_2} - v_{i-\frac{1}{2}\mathbf{e}_2}) \mathbf{e}_2 + \Delta x \Delta y (w_{i+\frac{1}{2}\mathbf{e}_3} - w_{i-\frac{1}{2}\mathbf{e}_3}) \mathbf{e}_3]$. The terms $F_{i\pm\frac{1}{2}\mathbf{e}_1}(\mathbf{u}, \phi)$ in Eq. (9) denote the numerical fluxes. These are discretized with a second order Godunov scheme for the case of momentum components [20], whereas the geometrical procedure indicated by Rider and Kothe [19] is used to calculate the f -fluxes.

2.2 Solution of the Discretized Poisson Problem

Because the flow is incompressible, an equation for pressure is needed to solve Eq. (2). This equation is obtained by applying the divergence-free condition of the velocity field to the time-discretized Eq. (2). One obtains:

$$\frac{\nabla \cdot \tilde{\mathbf{u}}}{\Delta t} = \frac{\nabla^2 p}{\rho}, \quad (10)$$

where $\tilde{\mathbf{u}} = \tilde{u}\mathbf{e}_1 + \tilde{v}\mathbf{e}_2 + \tilde{w}\mathbf{e}_3$ is an auxiliary velocity field where all acceleration terms of Eq. (2) but the pressure gradient have been added. By integrating Eq. (10) over a control volume Ω and using of the Gauss theorem one obtains:

$$\int_{\partial\Omega} \frac{\tilde{\mathbf{u}} \cdot \hat{\mathbf{n}}}{\Delta t} d\mathbf{x} = \int_{\partial\Omega} \frac{\nabla p \cdot \hat{\mathbf{n}}}{\rho} d\mathbf{x}, \quad (11)$$

where $\hat{\mathbf{n}}$ denotes the normal vector to the surface $\partial\Omega$. Discretization of Eq. 11 leads to a system of equations of the form:

$$\mathbf{A}\bar{p} = \mathbf{b}, \quad (12)$$

where \mathbf{A} is the system matrix with dimension $N_{cell} \times N_{cell}$, \bar{p} and \mathbf{b} are the vectors for the discrete p values and for the right hand side, respectively. Because the pressure gradient in Eq. (11) is discretized with central finite differences, on each line of the system corresponding to the control volume of index i all coefficients of \mathbf{A} are zero but for the ones with positions $S_A = \{0, \pm e_1, \pm e_2, \pm e_3\}$ with respect to i . S_A is called the structure of \mathbf{A} [21]. Equation (12) is solved by a multigrid solver embedded into *FS3D* which is specialized to deal with matrices which have the structure $S_A = \{0, \pm e_1, \pm e_2, \pm e_3\}$ for each computational cell.

3 Treatment of Embedded Boundaries

3.1 The Simplified Approach

To represent embedded boundaries, an additional volume fraction variable f_b is introduced and the boundary surface is also approximated with the PLIC scheme. Embedded boundaries are then treated as rigid bodies of infinite density. This is obtained by setting to zero all coefficients of Eq. (12), which correspond to momentum control volumes in which $f_b > 0$. This is equivalent to solving Eq. (12) for the *stair-step* approximation of the boundary (see Fig. 1a). As a consequence however, the velocities on all faces of the vast majority of the boundary-cut cells are set to zero. One a-posteriori treatment of the velocity field is then necessary to advect the fluid volume fraction in these cells. Boundary-cut cells with zero velocities on all their

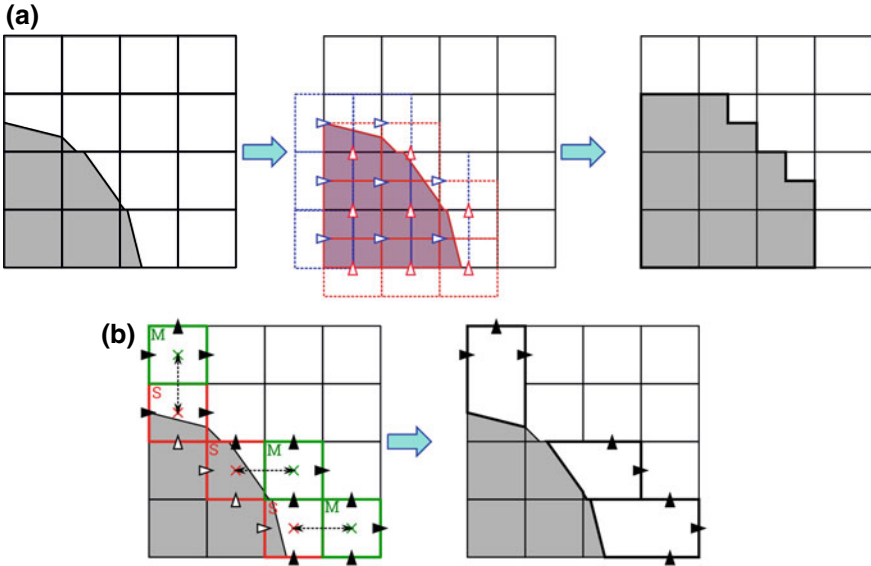


Fig. 1 The simplified approach to embed solid structures in a Cartesian grid. **a** Stair-step approximation of embedded boundaries. **b** Merging of critical cells with the neighbor in the direction of the largest normal component \hat{n}_b .

faces are marked as *slaves* and linked to the neighbor in the direction of the largest component of \hat{n}_b , which becomes their *master* (see Fig. 1b). This is made possible by the apposite structured data type *boundary cell*:

$$\mathbf{boundary\ cell} \begin{cases} \text{type}(\text{boundary cell}), \text{ pointer: } \text{master, slave} \\ \text{integer}(1:3) : & \mathbf{i} = ie_1 + je_2 + ke_3 \end{cases}$$

which is allocated in each boundary-cut cell and in its neighborhood. The advection of f is then carried out on the total volume of master-slave pairs using an averaged velocity field. The averaging of the velocity field after the solution of the Poisson equation causes an error in mass conservation. Even if those errors are very small in our simulations, this approach is fundamentally not conservative. With the purpose of improving conservativeness, we started the development of a new approach.

3.2 The New Approach

As in the simplified approach described in Sect. 3.1, embedded boundaries are represented by their volume fraction f_b and their surface is approximated with the PLIC scheme. However, instead of being treated as rigid bodies of infinite density, the boundaries are cut-off from the computational domain. Following the approach of

Popinet [15], the discretized equations for momentum transport (Eq. 2) and pressure (Eq. 10) are rewritten in terms of boundary-cut cell volume and faces. The same treatment is applied to the equations for interface transport (Eq. 5) and mass conservation (Eq. 1). Information about “free” volume and lateral “free” faces, as well as the orientation \mathbf{n}_b and position l^* of the plane approximating the surface must then be stored for each boundary-cut control volume. The structured data type *boundary cell* is then modified as follows:

$$\mathbf{boundary\ cell} \left\{ \begin{array}{ll} \text{real} : & l_b^*, f_b \\ \text{real}(1:3) : & \hat{\mathbf{n}}_b \\ \text{real}(1:3, 1:2) : & a_{bi} \\ \text{type}(\text{boundary cell}), \text{ pointer} : & \text{master, slave} \\ \text{integer}(1:3) : & \mathbf{i} = ie_1 + je_2 + ke_3 \end{array} \right.$$

where a_{bi} are the ratios of the lateral free to whole faces. For example:

$$a_{bi}(1, 1) \equiv a_{be_1}^- = \frac{A_{ie_1}^-}{A_{ie_1}}, \quad a_{bi}(1, 2) \equiv a_{be_1}^+ = \frac{A_{ie_1}^+}{A_{ie_1}}. \quad (13)$$

where $A_{ie_1}^\pm$ are the “free” lateral areas at positions $\mathbf{i} \pm \frac{1}{2}\mathbf{e}_1$. The *master* and *slave* pointer attribute are still needed to avoid problems caused by very small cut-cells, which we will address in Sect. 3.2.1. Since the grid is staggered, the structured data type *boundary cell* has to be allocated in staggered control volumes as well. Therefore, the new data type *boundary cell array*, is introduced:

$$\mathbf{boundary\ cell\ array} \left\{ \begin{array}{l} \text{type}(\text{boundary cell}) : bcell_c \\ \text{type}(\text{boundary cell}) : bcell_x \\ \text{type}(\text{boundary cell}) : bcell_y \\ \text{type}(\text{boundary cell}) : bcell_z \end{array} \right.$$

This data structure serves as container for *boundary cell* data types, one for each control volume associated with a position of index \mathbf{i} of the computational domain (one scalar control volume and three staggered control volumes). The *boundary cell array* is allocated in each position of the computational domain, in which at least one of the associated control volumes is intercepted by the boundary surface and in its neighborhood. Since phase change is not considered here, the treatment of the advection term is the point of interest for interface tracking (Eq. 5). In the case of momentum transport (Eq. 2), apart for advection and pressure, other terms have to be considered: the viscous stresses $\nabla \cdot \mathbb{S}$, the body forces $\rho \mathbf{g}$, and the surface tension force \mathbf{f}_σ . For the surface tension force no special treatment is needed, since it only depends on the interface orientation $\hat{\mathbf{n}}$. Body forces and viscous stresses are calculated by using material properties calculated on boundary-cut cell volumes. That is

$$\psi_i = \min \left(\left| \frac{f_i}{1 - f_{b,i}} \right|, 1 \right) \psi_l + \left(1 - \min \left(\left| \frac{f_i}{1 - f_{b,i}} \right|, 1 \right) \right) \psi_g, \quad (14)$$

where $0 < f_{b,i} < 1$ and ψ is a material property as density ρ or viscosity μ . Apart for this, the usual *FS3D* schemes for calculating viscous terms and body forces are used. This is equivalent to consider computational nodes on whole cell centres instead of boundary-cut cell centres. Of course, this simplification comes at the cost of slightly reduced accuracy, but it spares the computational overhead of calculating the barycentres of boundary-cut cells and the distance between the barycentres to compute gradients and linear interpolations of needed variables.

3.2.1 Advection in Boundary-Cut Cells

The split advection scheme, illustrated in Sect. 2.1, is also used on boundary-cut cells. Here Eq. (9) takes the form:

$$\begin{aligned} & \Omega(1 - f_{b,i}) \frac{\bar{\phi}_i^* - \bar{\phi}_i^n}{\Delta t} = \\ & -A_{e_1} (a_{be_1}^+ F_{i+\frac{1}{2}e_1}(\mathbf{u}, \phi) - a_{be_1}^- F_{i-\frac{1}{2}e_1}(\mathbf{u}, \phi)) \\ & + [(1 - \beta)\bar{\phi}_i^n + \beta\bar{\phi}_i^*](V_{i+\frac{1}{2}e_1} - V_{i-\frac{1}{2}e_1}), \end{aligned} \quad (15)$$

where $a_{be_1}^\pm$ are the free to whole area ratio at positions $i \pm \frac{1}{2}e_1$ and V denotes an altered form of the divergence correction. For momentum advection, $V_{i \pm \frac{1}{2}e_1} = A_{e_1} a_{be_1}^\pm \mathbf{u}_{i \pm \frac{1}{2}e_1}$, for f -advection, V is calculated geometrically and it corresponds to the ratio of the cut cell volume enclosed by $A_{e_1} \mathbf{u}_{i \pm \frac{1}{2}e_1} \Delta t$ in upwind control volumes to the time step Δt . *FS3D*'s usual schemes are used to compute the numerical fluxes. As already mentioned in Sect. 3.2, this is equivalent to considering computational nodes on whole cell centres and, consequently, faces. In the case of f -advection, the volume of arbitrary polyhedra has to be calculated in order to compute the fluxes. For this, geometrical algorithms similar to these proposed by Pathak and Raessi [22] are used. The interface position is calculated iteratively until the volume of the polyhedron representing the liquid phase corresponds to the liquid volume fraction f . As reported by Popinet [15], the occurrence of very small cells leads to prohibitively small time steps in order to satisfy the Courant-Friedrichs-Lewy (CFL) condition. Small cells are then marked as slaves and merged to the neighbor in the direction of the largest normal component, which becomes their master. Advection is then carried out in the whole master-slave control volume.

3.2.2 Discretization of the Poisson Problem on Boundary-Cut Cells

Discretization of Eq. (10) on boundary-cut volumes leads to:

$$\sum_k \left(A_{ab} \frac{\nabla p}{\rho} \right)_k = \frac{1}{\Delta t} \sum_k (A_{ab} u)_k \tag{16}$$

where $k = i \pm \frac{1}{2} \mathbf{e}_{1,2,3}$ are the positions of the faces of the considered scalar control volume. As already mentioned in Sects. 3.1 and 3.2.1, we consider computational nodes on whole face centres instead of boundary-cut cell centres. The pressure gradient $(\nabla p)_k$ can then be discretized by finite differences. Then, the formulation of Eq. (16) for each control volume leads to a system of equations analogously to Eq. (12). We observed that the use of whole-face centres may cause inaccurate values of the velocity field at very small boundary-cut cells. Better accuracy may be achieved by discretizing the pressure gradient differently, as indicated by Johansen and Colella [23]. However, this would change the structure of the system matrix \mathbf{A} , requiring a modification of the multigrid solver.

4 Results

In this section, a comparison of the two methods is shown for the case of a water drop impacting on the valley between two trapezoidal ribs at a Weber number $We = (\rho R U_0^2) / \sigma = 11.2$ (see Fig. 2). This setup is analogous to one of the cases studied by Regulagadda et al. [24]. Because of the symmetry of the problem, only a quarter of the computational domain was simulated with a total of $N_x \times N_y \times N_z = 192 \times 192 \times 192$ computational cells. The mesh was refined near the impact area.

In Fig. 3a comparison of the results obtained with the two methods is shown for different times during impact. It can be noted how the whole velocity field is affected by the choice of the method. Indeed, a difference is visible even before the drop touches the ribs. This is probably caused by the different treatment of the discretized elliptic Poisson equation. The impact occurs at a higher velocity in the new method, even if both simulations were initialized identically. Indeed, in Fig. 3b the droplet rim is thinner and faster and reaches a larger extension in the direction parallel to the ribs. The major differences however take place during the retraction phase. In both methods, the droplet assumes a three-lobed shape while retracting; in the approximate method however, these liquid sub-units merge just before take-off. The different

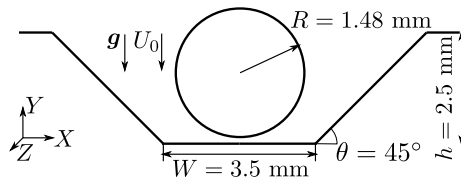


Fig. 2 Setup of the investigated case, which is analogous to one used in the experiments of Regulagadda et al. [24]

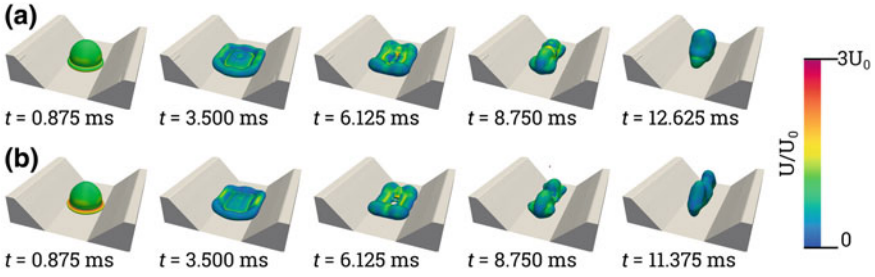


Fig. 3 Simulation results for drop impact on the valley between two trapezoidal ribs at $We = 11.2$. **a** Approximate approach. **b** New approach

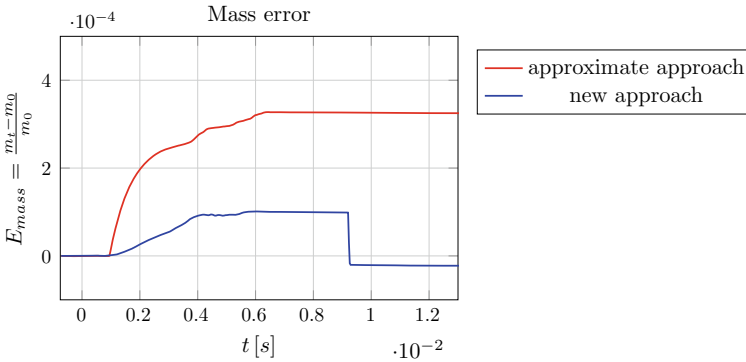


Fig. 4 Mass conservation error over time

impact morphology also explains the difference in contact time prediction. Indeed, according to Richard [25], the contact time t_0 on a flat superhydrophobic surface does not depend on impact velocity. Instead, it scales with the inertial-capillary time scale $\tau = \sqrt{\rho R^3 / \sigma}$. For the case of a droplet impacting on a wire, Gauthier et al. [2] found that the contact time t_c was approximately $t_0 / \sqrt{N_l}$, where N_l denotes the number of liquid sub-units formed during impact. The same trend was confirmed by Regulagadda et al. [3] for a drop impacting on the top and between triangular ribs. For the new approach, t_c / t_0 is closer to $1 / \sqrt{3}$, as expected from the number of liquid sub-units. The formation of three lobes was also observed by Regulagadda et al. for impact on the valley between two triangular ribs at higher impact velocities. Both methods however underestimate the contact time for this case, which is close to $2\tau = 13.316$ ms, as can be seen from a further publication from the same authors [24]. For both cases, we observe an un-physical rupture of the lamella at $t \approx 6$ ms, which in the new approach comes together with the ejection of a very small secondary droplet. This occurs probably because the grid used for the comparison is too large to capture the thin film on the top of the solid surface. With the new method, a significant improvement in mass conservation was obtained. In Fig. 4 the

mass error $E_m = (m(t) - m_0)/m_0$ is shown for both simulations as a function of time. The sudden step in the mass error for the new approach at $t \approx 9.2$ ms is due to the ejection of the satellite droplet. It can be seen that the error in mass conservation could be reduced of about a factor of 3. We have indeed $E_{m, \infty, old} = \|(m(t) - m_0)/m_0\|_{\infty} = 3.27 \times 10^{-4}$ for the approximate approach and $E_{m, \infty, new} = 1.01 \times 10^{-4}$ for the new one. This is a remarkable achievement considering the fact that the contact time between liquid and ribs is short. Better performances are expected for cases in which the interaction between droplet and solid structures has a longer duration. Further investigation on the causes of the residual error in mass conservation is planned.

5 Conclusions

Two methods to embed boundaries of arbitrary shape in a Cartesian grid have been presented, one approximate and a more rigorous one. A comparison between the two approaches for the case of a droplet impacting between two trapezoidal ribs has shown a considerable difference in impact dynamics. The cause of this discrepancy still needs to be clarified, but we expect it to be due to the different treatment of the discretized Poisson problem. Even if the newer approach has shown a slightly larger discrepancy for the contact time prediction for the case under consideration, further research is needed to assess its accuracy for other surface geometries and impact velocities. On the other hand, a remarkable reduction of the mass conservation error was obtained. Small error in mass conservation are still present however, the cause of which will be addressed by further research.

Acknowledgements We thank the German Science Foundation (DFG) for the financial support of this research within the international research training group *Droplet Interaction Technologies, DROPIT*, GRK 2016/1.

References

1. Bird, J.C., Dhiman, R., Kwon, H., Varanasi, K.K.: Reducing the contact time of a bouncing drop. *Nature* **503**, 385–388 (2013)
2. Gauthier, A., Symon, S., Clanet, C., Quéré, D.: Water impacting on superhydrophobic macro-textures. *Nat. Commun.* **6**, 8001 (2015)
3. Regulagadda, K., Bakshi, S., Das, S.K.: Morphology of drop impact on a superhydrophobic surface with macro-structures. *Phys. Fluids* **29**, 082104 (2017)
4. Chantelot, P., Moqaddam, A.M., Gauthier, A., Chikatamaria, S.S., Clanet, C., Karlin, I.V., Quéré, D.: Water ring-bouncing on repellent singularities. *Soft. Matter* **14**(12), 2227–2233 (2018)
5. Shen, Y., Liu, S., Zhu, C., Tao, J., Chen, Z., Tao, H., Pan, L., Wang, G., Wang, T.: Bouncing dynamics of impact droplets on the convex superhydrophobic surfaces. *Appl. Phys. Lett.* **110**, 221601 (2017)

6. Khojasteh, D., Bordbar, A., Kamali, R., Marengo, M.: Curvature effect on droplet impacting onto hydrophobic and superhydrophobic spheres. *Int. J. Comput. Fluid Dyn.* **31**(6–8), 310–323 (2017)
7. Liu, Y., Andrew, M., Li, J., Yeomans, J.M., Wang, Z.: Symmetry breaking in drop bouncing on curved surfaces. *Nat. Commun.* **6**, 10034 (2015)
8. Liu, X., Zhao, Y., Chen, S., Shen, S., Zhao, X.: Numerical research on the dynamic characteristics of a droplet impacting a hydrophobic tube. *Phys. Fluids* **29**, 062105 (2017)
9. Schlottke, J., Dulger, E., Weigand, B.: A VOF-based 3D numerical investigation of evaporating, deformed droplets. *Prog. Comput. Fluid Dyn. Int. J.* **9**, 426–435 (2009)
10. Schlottke, J., Straub, W., Beheng, K.D., Gomma, H., Weigand, B.: Numerical investigation of collision-induced breakup of raindrops. Part I: methodology 12 references and dependencies on collision energy and eccentricity. *J. Atmos. Sci.* **67**, 557–575 (2010)
11. Ertl, M., Weigand, B.: Analysis methods for direct numerical simulations of primary breakup of shear-thinning liquid jets. *Atomization Sprays* **27**(4), 303–317 (2017)
12. Reitzle, M., Kieffer-Roth, C., Garcke, H., Weigand, B.: A volume-of-fluid method for three-dimensional hexagonal solidification processes. *J. Comput. Phys.* **339**, 356–369 (2017)
13. Reitzle, M., Ruberto, S., Stierle, R., Gross, J., Tanzen, T., Weigand, B.: Direct numerical simulation of sublimating ice particles. *Int. J. Thermal Sci.* **145**, 105953 (2019)
14. Rauschenberger, P., Weigand, B.: Direct numerical simulation of rigid bodies in multiphase flow within an Eulerian framework. *J. Comput. Phys.* **291**, 238–253 (2015)
15. Popinet, S.: Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries. *J. Comput. Phys.* **190**(2), 572–600 (2003)
16. Eisenschmidt, K., Ertl, M., Gomma, H., Kieffer-Roth, C., Meister, C., Rauschenberger, P., Reitzle, M., Schlottke, K., Weigand, B.: Direct numerical simulations for multiphase flows: an overview of the multiphase code FS3D. *Appl. Math. Comput.* **272**, 508–517 (2016)
17. Lafaurie, B., Nardone, C., Scardovelli, R., Zaleski, S., Zanetti, G.: Modelling merging and fragmentation in multiphase flows with SURFER. *J. Comput. Phys.* **113**(1), 134–147 (1994)
18. Hirt, C.W., Nichols, B.D.: Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comput. Phys.* **39**, 201–225 (1981)
19. Rider, W.J., Kothe, D.B.: Reconstructing volume tracking. *J. Comput. Phys.* **141**(2), 112–152 (1998)
20. Rieber, M.: Numerische Modellierung der Dynamik freier Grenzflächen in Zweiphasenströmungen, dissertation, University of Stuttgart (2004)
21. Wesseling, P.: *An Introduction to Multigrid Methods*. Wiley (1992)
22. Pathak, A., Raessi, M.: A three-dimensional volume-of-fluid method for reconstructing and advecting three-material interfaces forming contact lines. *J. Comput. Phys.* **307**, 550–573 (2016)
23. Johansen, H., Colella, P.: A Cartesian grid embedded boundary method for Poisson's equation on irregular domains. *J. Comput. Phys.* **147**(1), 60–85 (1998)
24. Regulagadda, K., Bakshi, S., Das, S.K.: Triggering of flow asymmetry by anisotropic deflection of lamella during the impact of a drop onto superhydrophobic surfaces. *Phys. Fluids* **30**, 072105 (2018)
25. Richard, D., Clanet, C., Quéré, D.: Contact time of a bouncing drop. *Nature* **417**, 811 (2002)