INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2016; **80**:53–75
Published online 24 July 2015 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/fld.4071

Numerical simulations of bouncing jets

Andrea Bonito*,†, Jean-Luc Guermond and Sanghyun Lee

Department of Mathematics, Texas A&M University, College Station, Texas 77843-3368, USA

SUMMARY

Bouncing jets are fascinating phenomenon occurring under certain conditions when a jet impinges on a free surface. This effect is observed when the fluid is Newtonian and the jet falls in a bath undergoing a solid motion. It occurs also for non-Newtonian fluids when the jets fall in a vessel at rest containing the same fluid. We investigate numerically the impact of the experimental setting and the rheological properties of the fluid on the onset of the bouncing phenomenon. Our investigations show that the occurrence of a thin lubricating layer of air separating the jet and the rest of the liquid is a key factor for the bouncing of the jet to happen. The numerical technique that is used consists of a projection method for the Navier–Stokes system coupled with a level set formulation for the representation of the interface. The space approximation is carried out with adaptive finite elements. Adaptive refinement is shown to be very important to capture the thin layer of air that is responsible for the bouncing. Copyright © 2015 John Wiley & Sons, Ltd.

Received 16 February 2015; Revised 2 June 2015; Accepted 27 June 2015

KEY WORDS: bouncing jet; Kaye effect; entropy viscosity; level set; projection method; shear-thinning viscosity; adaptive finite elements

1. INTRODUCTION

The ability of a jet of fluid to bounce on a free surface has been observed in different contexts. Thrasher *et al*. [1] designed an experiment where a jet of Newtonian fluid falls into a rotating vessel filled with the same fluid. The authors investigated the conditions under which the jet bounces: nature of the fluid, jet diameter, and jet and bath velocities. We refer to [2] for experimental movies illustrating this effect. Bouncing can also be observed in a stationary vessel provided the jet is composed of a *non-Newtonian* fluid. During the pouring process, a small heap of fluid forms and the jet occasionally leaps upward from the heap (Figure 1). This is the so-called *Kaye* effect as first observed by [3] in 1963. About 13 years after this phenomenon was first mentioned in the literature, [4] revisited the experiment and suggested that the ability for the fluid to exhibit shear-thinning viscosity and elastic behavior are key ingredients for the bouncing to occur. Additional laboratory experiments performed by [5] and [6] lead the authors of each team to propose a list of properties that the fluid should have for the Kaye effect to occur. The conclusions of these two papers disagree on the requirement that the fluid be elastic and on the nature of the thin layer that separates the heap and the outgoing jet. It is argued in [5] that elastic properties are not necessary and that the thin layer is a shear layer, whereas it is argued in [6] that the fluid should have elastic properties and that the thin layer separating the heap from the bouncing jet is a layer of air. Recent experiments reported in [7] using a high speed camera unambiguously show that the jet slides on a lubricating layer of air. For completeness, we also refer to [8] for a thorough discussion on the 'stable' Kaye effect, where the jet falls against an inclined surface.

*Correspondence to: Andrea Bonito, Department of Mathematics, Texas A&M University, College Station, Texas 77843-3368, USA.

†E-mail: bonito@math.tamu.edu

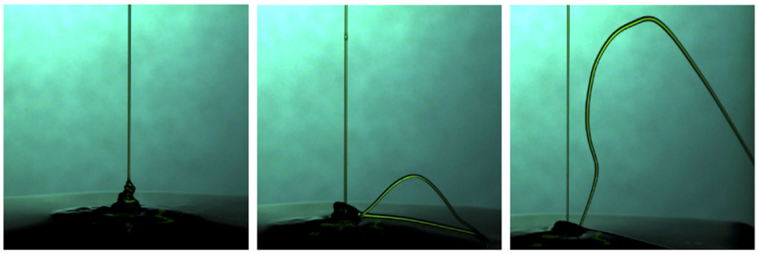


Figure 1. Experimental observation of the Kaye effect. (Left) The fluid starts to buckle producing a heap; (middle) a stream of liquid suddenly leaps outside the heap; (right) fully developed Kaye effect.

The objective of the present paper is to numerically revisit the Kaye effect. Our key finding is that *a thin air layer is always present between the bouncing jet and the rest of the fluid whether the fluid is Newtonian or not.* Our numerical experiments suggest that the critical parameter for bouncing to occur is that the properties of fluid and the experimental conditions be such that a stable layer of air separating the jet and the ambient fluid can appear. This condition is met by setting the bath in motion for Newtonian fluids; it can also be met if the bath is stationary provided that the fluid is non-Newtonian and has shear-thinning viscosity. The numerical code that we use is based on a modeling of the fluid motion by the incompressible Navier–Stokes equations supplemented with a surface tension mechanism. The shear-thinning viscosity of the non-Newtonian fluids is assumed to follow a model by [9]. Elastic behaviors are not modeled. The numerical approximation of the resulting two-phase flow model is based on two solvers: one solving the Navier–Stokes equations assuming that the fluid/air distribution is given, and the other keeping track of the motion of the interface assuming that the transport velocity is given. The Navier–Stokes solver is based on a version of the projection method of [10] and [11] using a novel adaptive in time second-order backward differentiation formula (BDF2) for the time discretization and adaptive finite elements for the space approximation. The transport solver is based on a level set technique in the spirit of [12] to represent the liquid/air interface. The level set is approximated in space by using adaptive finite elements to accurately capture the aforementioned thin air layer, and the adaptive time stepping is carried out by using a third-order explicit Runge–Kutta technique. The transport of the level set function is stabilized using a novel 'on the fly' entropy residual method mathematically studied in [13] in a model setting, which guarantees an accurate representation of the free boundary.

The paper is organized as follows. The mathematical model is presented in Section 2. The numerical techniques to solve the Navier–Stokes equations and the transport equation for the level set function are described in Section 3. Various validation tests of the numerical algorithms and comparisons with classical benchmark problems are reported in Section 4. Finally, we report numerical evidences of Newtonian and non-Newtonian bouncing jets in Section 5.

2. THE MATHEMATICAL MODEL

This section presents the mathematical models adopted to describe non-mixing two-phase fluid flows with capillary forces. Each fluid is assumed to be incompressible.

2.1. *Two-phase flow system*

Let $\Lambda \subset \mathbb{R}^d (d = 2, 3)$ be an open and bounded computational domain with Lipschitz boundary $\partial\Lambda$ and let $[0, T]$ be the computational time interval, $T > 0$. The cavity Λ is filled with two non-mixing fluids undergoing some time-dependent motion, say fluid 1 and fluid 2. We denote by Ω^+ and Ω^- the open subsets of the space-time domain $\Lambda\times[0, T]$ occupied by fluid 1 and fluid 2, respectively. We denote by $\rho^+, \mu^+, \rho^-, \mu^-$ the density and dynamical viscosity of each fluid, respectively. The interface between the two fluids in the space-time domain is denoted $\Sigma := \partial\Omega^+ \cap \partial\Omega^-$, and the normal to Σ, oriented from Ω^+ to Ω^-, is denoted $\mathbf{n}_\Sigma$. The two space-time components of the vector field $\mathbf{n}_\Sigma : \Sigma \longrightarrow \mathbb{R}^d \times \mathbb{R}$ are denoted $\mathbf{n}$ and $\mathbf{n}_\tau$, respectively. It is also useful to define $\Omega^\pm(t) := \Omega^\pm \cap (\Lambda \times \{t\})$; that is, the sets $\Omega^+(t)$ and $\Omega^-(t)$ are the regions occupied by fluid 1 and

DOI: 10.1002/fld

fluid 2 at time t, respectively. We also introduce the interface $\Sigma(t) = \partial\Omega^+(t) \cap \partial\Omega^-(t)$; note that $\mathbf{n}$, as defined earlier, is the unit normal of $\Sigma(t)$ and it is oriented from $\Omega^+(t)$ to $\Omega^-(t)$. To facilitate the modeling, we define global density and dynamical viscosity functions $\rho, \mu : \Lambda \times [0,T] \to \mathbb{R}$ by setting $\rho(\mathbf{x},t) = \rho^\pm$ and $\mu(\mathbf{x},t) = \mu^\pm$ if $(\mathbf{x},t) \in \Omega^\pm$. The fluid velocity field $\mathbf{u} : \Lambda \times [0,T] \to \mathbb{R}^d$, henceforth assumed to be continuous across Σ, and the pressure $p : \Lambda \times [0,T] \to \mathbb{R}$ are defined globally and solve the incompressible Navier–Stokes equations in the distribution sense in the space-time domain

$$\rho\left(\frac{\partial}{\partial t}\mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u}\right) - 2\mathrm{div}\left(\mu\nabla^S\mathbf{u}\right) + \nabla p - \delta_\Sigma\sigma\kappa\mathbf{n} = \rho\mathbf{g} \qquad \text{in } \Lambda \times (0,T], \tag{1a}$$

$$\mathrm{div}(\mathbf{u}) = 0 \qquad \text{in } \Lambda \times (0,T], \tag{1b}$$

where $\nabla^S\mathbf{u} := \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^T)$ is the strain rate tensor, $\mathbf{g}$ is the gravity field, and $\delta_\Sigma\sigma\kappa\mathbf{n}$ is a singular measure modeling the surface tension acting on $\Sigma(t)$. The distribution δ_Σ is the Dirac measure supported on Σ, the function $\kappa : \Sigma \to \mathbb{R}$ is the total curvature of $\Sigma(t)$ (sum of the principal curvatures) and σ is the surface tension coefficient.

The system (1) is supplemented with initial and boundary conditions. The initial condition is $\mathbf{u}(\mathbf{x},0) = \mathbf{u}_0(\mathbf{x})$ for all $\mathbf{x} \in \Lambda$, where $\mathbf{u}_0$ is assumed to be a smooth divergence-free velocity field. The boundary $\partial\Lambda$ is decomposed into three non-overlapping components $\partial\Lambda := \overline{\Gamma}_D \cup \overline{\Gamma}_N \cup \overline{\Gamma}_S$ with $\Gamma_D \cap \Gamma_N = \emptyset, \Gamma_N \cap \Gamma_S = \emptyset, \Gamma_S \cap \Gamma_D = \emptyset$. Time-dependent decomposition could be considered but are not described here to avoid unnecessary technicalities. Given $\mathbf{f}_N : \Gamma_N \to \mathbb{R}^d$ and $\mathbf{f}_D : \Gamma_D \to \mathbb{R}^d$, we require that

$$\left(2\mu\nabla^S\mathbf{u} - pI\right)\boldsymbol{\nu} = \mathbf{f}_N, \qquad \text{on } \Gamma_N \times (0,T], \tag{2a}$$

$$\mathbf{u} = \mathbf{f}_D, \qquad \text{on } \Gamma_D \times (0,T], \tag{2b}$$

$$\mathbf{u}\cdot\boldsymbol{\nu} = 0, \quad \left(\left(2\mu\nabla^S\mathbf{u} - pI\right)\boldsymbol{\nu}\right)\times\boldsymbol{\nu} = 0, \qquad \text{on } \Gamma_S \times (0,T], \tag{2c}$$

where $\boldsymbol{\nu}$ is the outward pointing unit normal on $\partial\Lambda$ and I is the $d \times d$ identity matrix. For simplicity, we assume that the $(d-1)$ measures of $\Gamma_N \cup \Gamma_S$ and Γ_D are each strictly positive; otherwise, extra constraints either on the velocity or on the pressure must be enforced.

Note that we could have formulated the conservation of momentum without invoking the singular measure modeling the surface tension by saying that (1a) holds in Ω^+ and Ω^- (without the singular measure) and by additionally requiring that

$$[\mathbf{u}] = 0 \quad \text{and} \quad \left[2\mu\nabla^S\mathbf{u} - pI\right]\mathbf{n} = \sigma\kappa\mathbf{n} \quad \text{on} \quad \Sigma(t), \qquad \forall t \in (0,T], \tag{3}$$

where $[.]$ denotes the jump across $\Sigma(t)$ defined by $[\mathbf{v}](\mathbf{x}) := \lim_{\Omega^-\ni\mathbf{y}\to\mathbf{x}}\mathbf{v}(\mathbf{y}) - \lim_{\Omega^+\ni\mathbf{z}\to\mathbf{x}}\mathbf{v}(\mathbf{z})$, that is, $[\mathbf{v}](\mathbf{x}) = \mathbf{v}(\mathbf{x}^-) - \mathbf{v}(\mathbf{x}^+)$, for all $\mathbf{x} \in \Sigma(t)$ and all $\mathbf{v} : \Lambda \to \mathbb{R}^d$ or $\mathbf{v} : \Lambda \to \mathbb{R}^{d\times d}$.

The interface $\Sigma(t)$ is assumed to be transported by the fluid particles. More precisely, let $\partial^-(\Lambda\times[0,T])$ be the part of the boundary of the space-time domain where the characteristics generated by the field $(\mathbf{u},1)$ enter, that is,

$$\partial^-(\Lambda \times [0,T]) := \Lambda \times \{0\} \cup \{(\mathbf{x},t) \in \partial\Lambda \times [0,T] \,|\, \mathbf{u}(\mathbf{x},t)\cdot\boldsymbol{\nu} < 0\}. \tag{4}$$

We then define $\{\mathbf{x}(\mathbf{P},t) \in \Lambda,\ t \in [t_\mathbf{P},T],\ (\mathbf{P},t_\mathbf{P}) \in \partial^-(\Lambda\times[0,T])\}$ to be the family of the characteristics generated by the velocity field $\mathbf{u}$, that is, $\frac{\partial}{\partial t}\mathbf{x}(\mathbf{P},t) = \mathbf{u}(\mathbf{x}(\mathbf{P},t),t)$ with $\mathbf{x}(\mathbf{P},t_\mathbf{P}) = \mathbf{P}$, $(\mathbf{P},t_\mathbf{P}) \in \partial^-(\Lambda\times[0,T])$ where $t_\mathbf{P}$ is the time when the characteristics enters the space-time domain $\Lambda\times[0,T]$ at $\mathbf{P}$. Let us now denote by $\Sigma_0 := \Sigma \cap \partial^-(\Lambda\times[0,T])$ the location of the interface at the inflow boundary of the space-time domain, then we are going to assume in the entire paper that the velocity field is smooth enough so that the following property holds

$$\forall\mathbf{x} \in \Sigma(t),\ \exists!(\mathbf{P},t_\mathbf{P}) \in \Sigma_0,\ \mathbf{x} = \mathbf{x}(\mathbf{P},t),\ t \geqslant t_\mathbf{P}. \tag{5}$$

DOI: 10.1002/fld

2.2. *Eulerian representation of the free boundary interface*

A level set technique is used to keep track of the position of the time-dependent interface $\Sigma(t)$, see for instance [12]. This method is recalled in Section 2.2.1. A typical problem arising when using a level set method to describe interfaces is to guarantee the non-degeneracy of the representation. We discuss a reinitialization technique overcoming this issue in Section 2.2.2.

2.2.1. Level set representation. Let us define the so-called level function $\phi(\mathbf{x},t) : \Lambda \times [0,T] \to \mathbb{R}$ so that

$$\frac{\partial}{\partial t}\phi + \mathbf{u}\cdot\nabla\phi = 0 \quad \text{in} \quad \Lambda \times (0,T], \qquad \phi(\mathbf{P},t_{\mathbf{P}}) = \phi_0(\mathbf{P},t_{\mathbf{P}}) \quad \text{on} \quad \partial^-(\Lambda \times [0,T]), \tag{6}$$

where we assume that ϕ_0 is a smooth function satisfying the following properties:

$$\partial\Omega^{\pm} \cap \partial^-(\Lambda \times [0,T]) = \{(\mathbf{P},t_{\mathbf{P}}) \in \partial^-(\Lambda \times [0,T]) \mid \pm\phi_0(\mathbf{P},t_{\mathbf{P}}) \geqslant 0\}, \tag{7}$$

$$\Sigma_0 := \Sigma \cap \partial^-(\Lambda \times [0,T]) = \{(\mathbf{P},t_{\mathbf{P}}) \in \partial^-(\Lambda \times [0,T]) \mid \phi_0(\mathbf{P},t_{\mathbf{P}}) = 0\}. \tag{8}$$

Note that this definition implies that Σ_0 is the zero level set of ϕ_0. Upon introducing $\psi(\mathbf{P},t) := \phi(\mathbf{x}(\mathbf{P},t),t)$ for all $(\mathbf{P},t_{\mathbf{P}}) \in \partial^-(\Lambda \times [0,T]), t \geqslant t_{\mathbf{P}}$, the definition of ϕ together with the definition of the characteristics implies that $\partial_t\psi(\mathbf{P},t) = 0$ thereby proving that $\phi(\mathbf{x}(\mathbf{P},t),t) = \psi(\mathbf{P},t) = \psi(\mathbf{P},t_{\mathbf{P}}) = \phi(\mathbf{x}(\mathbf{P},t_{\mathbf{P}}),t_{\mathbf{P}}) = \phi(\mathbf{P},t_{\mathbf{P}}) = \phi_0(\mathbf{P},t_{\mathbf{P}})$. This means that the value of $\phi(\mathbf{x}(\mathbf{P},t),t)$ along the trajectory $\{\mathbf{x}(\mathbf{P},t_{\mathbf{P}}),\ t \in [t_{\mathbf{P}},T]\}$ is constant; in particular, the sign of $\phi(\mathbf{x}(\mathbf{P},t),t)$ does not change. This leads us to adopt the following alternative definitions for $\Omega^{\pm}$ and $\Sigma(t)$:

$$\Omega^{\pm} := \{(\mathbf{x},t) \in \Lambda\times[0,T] \mid \pm\phi(\mathbf{x},t) > 0\}, \tag{9a}$$

$$\Sigma(t) := \{\mathbf{x} \in \Lambda \mid \phi(\mathbf{x},t) = 0\}. \tag{9b}$$

The earlier characterizations will be used in the rest of the paper.

To simplify the presentation, we are going to assume in the rest of the paper that there is $\Gamma_L \subset \partial\Lambda$ such that $\partial^-(\Lambda \times (0,T)) = \Gamma_L \times (0,T)$, that is, the inflow boundary for the level set equation is time independent. We then set $\phi_L(\mathbf{P},t) = \phi_0(\mathbf{P},t)$ for all $\mathbf{P} \in \Gamma_L$, $t \in [0,T]$, and we abuse the notation by setting $\phi_0(\mathbf{P}) = \phi_0(\mathbf{P},0)$ for all $\mathbf{P} \in \Lambda$.

2.2.2. Reinitialization and cut-off function. A typical issue when dealing with level set representation of interfaces is to guarantee that the manifold $\{\mathbf{x} \in \Lambda \mid \phi(\mathbf{x},t) = 0\}$ is $(d-1)$-dimensional, that is, we want to make sure that $\|\nabla\phi\|_{\ell^2} > 0$ in every small neighborhood of the zero level set, where $\|\cdot\|_{\ell^2}$ is the Euclidean norm in $\mathbb{R}^d$. In order to achieve this objective, we implement an 'on the fly' reinitialization algorithm proposed in [14], which consists of replacing (6) by

$$\frac{\partial}{\partial t}\phi_{\lambda,\beta} + \mathbf{u}\cdot\nabla\phi_{\lambda,\beta} = \lambda\mathrm{sign}(\phi_{\lambda,\beta})\left(\mathcal{G}(\phi_{\lambda,\beta}) - \|\nabla\phi_{\lambda,\beta}\|_{\ell^2}\right), \tag{10}$$

where $\lambda, \beta > 0$ are parameters yet to be defined and $\mathcal{G}(\cdot)$, $\mathrm{sign}(\cdot)$ are defined by

$$\mathcal{G}(z) := 1 - \left(\frac{z}{\beta}\right)^2, \qquad \mathrm{sign}(z) := \begin{cases} 1, & z > 0 \\ -1, & z < 0 \\ 0, & |z| = 0. \end{cases} \tag{11}$$

The rationale for the new definition (10) is that the presence of the sign function in the right-hand side implies that the zero level set of $\phi_{\lambda,\beta}$ is the same as that of ϕ; that is, the characterizations of $\Omega^{\pm}$ and $\Sigma(t)$ are unchanged (9). Moreover, assuming that $\mathbf{u}$ is locally the velocity of a solid motion and upon setting $\psi(\mathbf{P},t) = \phi_{\lambda,\beta}(\mathbf{x}(\mathbf{P},t))$, we have $\frac{\partial}{\partial t}\psi = \lambda\mathrm{sign}(\psi)\left(\left(1-\left(\frac{\psi}{\beta}\right)^2\right) - |\nabla\psi|\right)$.

Assuming that this eikonal equation has a steady-state solution ψ_∞ and denoting by Σ_∞ the zero-level set of this steady-state solution, the behavior of ψ_∞ in the vicinity of Σ_∞ is $\psi_\infty(\mathbf{P}) \approx \beta \tanh\left(\frac{\mathrm{dist}(\Sigma_\infty,\mathbf{P})}{\beta}\right)$, because the solution to the following ODE : $y'(z) = 1 - (y(z)/\beta)^2$ is

$$\mathcal{F}(z) := \beta \tanh\left(\frac{z}{\beta}\right). \tag{12}$$

Note that ψ is close to ψ_∞ if λ is very large. In conclusion, the solution of (10) is such that $\phi_{\lambda,\beta}(\mathbf{x},t) \approx \beta \tanh\left(\frac{\mathrm{dist}(\Sigma(t),\mathbf{x})}{\beta}\right)$ if λ is large enough, that is, $\|\nabla\phi\|_{\ell^2} \sim 1$ in any small neighborhood of the zero level set. We are going to abuse the notation in the rest of the paper by dropping the indices λ, β and by using ϕ instead of $\phi_{\lambda,\beta}$.

3. NUMERICAL METHOD

We discuss the approximations of the level set equation in Section 3.1 and that of the Navier–Stokes system in Section 3.2. We consider a mesh sequence, $\{\mathcal{T}_h\}_{h>0}$, and we assume that each mesh $\mathcal{T}_h$ is a subdivision of $\overline{\Lambda}$ made of disjoint elements K, that is, rectangles when $d = 2$ or cuboids when $d = 3$. We denote by $\mathcal{E}(\mathcal{T}_h)$ the collection of interfaces and boundary faces (edge when $d = 2$ and faces when $d = 3$). Each subdivision is assumed to exactly approximate the computational domain, that is, $\overline{\Lambda} = \cup_{K\in\mathcal{T}_h} K$, and to be consistent with the decomposition of the boundary, that is, there exists $\mathcal{E}_I(\mathcal{T}_h) \subset \mathcal{E}(\mathcal{T}_h)$ for $I \in \{D, N, S, L\}$ such that $\overline{\Gamma}_I = \cup_{F\in\mathcal{E}_I(\mathcal{T})} F$. The diameter of an element $K \in \mathcal{T}_h$ is denoted by h_K; $h_K := \max_{\mathbf{x},\mathbf{y}\in K} |\mathbf{x}-\mathbf{y}|$. The mesh sequence $\{\mathcal{T}_h\}_{h>0}$ is assumed to be shape regular in the sense of Ciarlet. For any integer $k \geqslant 1$ and any $K \in \mathcal{T}_h$, we denote by $\mathbb{Q}^k(K)$ the space of scalar-valued multivariate polynomials over K of partial degree at most k. The vector-valued version of $\mathbb{Q}^k(K)$ is denoted $\mathbf{Q}^k(K)$. The index h is dropped in the rest of the paper, and we write $\mathcal{T}$ instead of $\mathcal{T}_h$ when the context is unambiguous.

Regarding the time discretization, given an integer $N \geqslant 2$, we define a partition of the time interval $0 =: t^0 < t^1 < \cdots < t^N := T$ and denote $\delta t^n := t^n - t^{n-1}$ and $t^{n+\frac{1}{2}} := t^n + \frac{1}{2}\delta t^{n+1}$.

3.1. Numerical approximation of the level set system

The continuous finite element method used for the space approximation of level set Equation (10) is described in Section 3.1.1. We present in Sections 3.1.2 an entropy-residual technique that has the advantage of avoiding the spurious oscillations that would otherwise be generated by using an unstabilized Galerkin technique. The time stepping is performed by using an explicit strong stability preserving (SSP) Runge–Kutta 3 (RK3) scheme as explained in Section 3.1.3.

3.1.1. Approximation in space. The space approximation of $\Phi(\cdot,t)$ of the level set function $\phi(\cdot,t)$ solution to (10) is carried out by using continuous, piecewise linear polynomials subordinate to the subdivision $\mathcal{T}$. The associated finite element spaces are defined by

$$\mathbb{W}(\mathcal{T}) := \{W \in C^0(\overline{\Lambda};\mathbb{R}) \mid W|_K \in \mathbb{Q}^1(K),\ \forall K \in \mathcal{T}\}, \tag{13}$$

$$\mathbb{W}_0(\mathcal{T}) := \{W \in C^0(\overline{\Lambda};\mathbb{R}) \mid W = 0 \text{ on } \Gamma_L,\ W|_K \in \mathbb{Q}^1(K),\ \forall K \in \mathcal{T}\}, \tag{14}$$

$$\mathbb{W}_L(\mathcal{T}) := \{W \in C^0(\overline{\Lambda};\mathbb{R}) \mid W = \Phi_L \text{ on } \Gamma_L,\ W|_K \in \mathbb{Q}^1(K),\ \forall K \in \mathcal{T}\}, \tag{15}$$

where $\Phi_L(\cdot,t)$ is a piecewise linear approximation of the inflow data $\phi_L(\cdot,t)$. Assuming that the velocity field $\mathbf{u} : \Lambda \times [0,T] \longrightarrow \mathbb{R}^d$ is known, the Galerkin approximation of (10) is formulated as follows: Given Φ_L and $\Phi(\cdot,0) = \Phi_0$, where $\Phi_0 \in \mathbb{W}(\mathcal{T})$ is an approximation of the initial condition ϕ_0, find $\Phi \in \mathcal{C}^1([0,T];\mathbb{W}_L(\mathcal{T}))$ such that

$$\int_\Lambda \frac{\partial}{\partial t}\Phi W = -\int_\Lambda \mathbf{u}\cdot\nabla\Phi\, W + \int_\Lambda \lambda\,\mathrm{sign}(\Phi)\,(\mathcal{G}(\Phi) - \|\nabla\Phi\|_{\ell^2})\, W, \quad \forall W \in \mathbb{W}_0(\mathcal{T}). \tag{16}$$

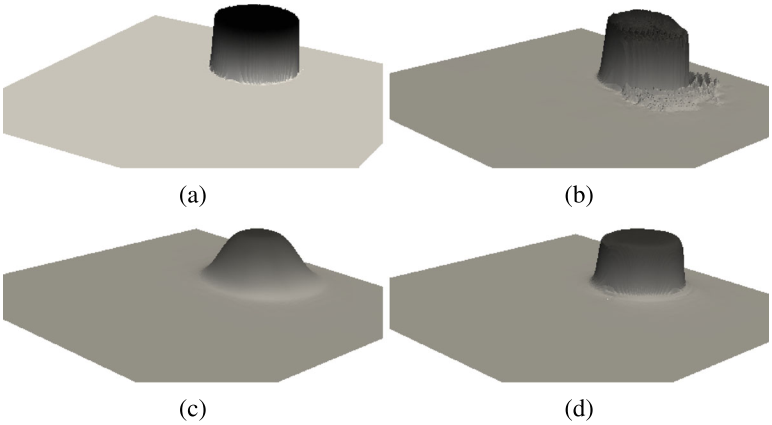


Figure 2. Graph of the level set function in the computational domain $\Lambda = (-1,1)^2$ using the velocity given in (48) : (a) initial data: ϕ_0 is the characteristic function of the disk of radius 1 centered at $\left(\frac{1}{2},0\right)$; (b) no stabilization; (c) first-order stabilization with $C_{\text{Lin}} = 0.1$; and (d) entropy viscosity stabilization with $C_{\text{Lin}} = 0.1$ and $C_{\text{Ent}} = 0.1$. Observe the spurious oscillations in panel (b) when no stabilization is applied. Both the first-order and the entropy viscosity solutions are free of oscillations, the latter being clearly more accurate.

It is well known that the solution to the preceding system exhibits spurious oscillations in the regions where $\|\nabla\Phi\|_{\ell^2}$ is large. We address this issue in the next section.

3.1.2. Entropy residual stabilization. We describe in this section an entropy viscosity technique to stabilize the Galerkin formulation (16). This method has been introduced in [15], and we refer to [13] for a mathematical discussion on its stability properties. To motivate the discussion, we refer to the panel (b) of Figure 2 showing the Galerkin approximation of the characteristic function of the unit disk, initially centered at $(\frac{1}{2},0)$, after one rotation about the origin.

The spurious oscillations are avoided by augmenting (16) with an artificial viscosity term where the viscosity is localized and chosen to be proportional to an entropy residual. To describe the method and define an appropriate local 'viscosity', we recall that the following holds in the distribution sense for any $E \in \mathcal{C}^1(\Lambda;\mathbb{R})$:

$$\frac{\partial}{\partial t}E(\phi) + \mathbf{u}\cdot\nabla E(\phi) - \lambda \text{sign}(\phi)\left(\mathcal{G}(\Phi) - \|\nabla\phi\|_{\ell^2}\right)E'(\phi) = 0,$$

Consequently, it is reasonable to expect that the semi-discrete entropy residual

$$R^{\text{Ent}}(\Phi,\mathbf{u}) := \frac{\partial}{\partial t}E(\Phi) + \mathbf{u}(t)\cdot\nabla E(\Phi) - \lambda \text{sign}(\Phi)\left(\mathcal{G}(\Phi) - \|\nabla\Phi\|_{\ell^2}\right)E'(\Phi)$$

is a reliable indicator of the regularity of ϕ. This quantity should be of the order of the consistency error in the regions where ϕ is smooth, and it should be large in the region where the partial differential equation is not well solved. In our computations, we have chosen

$$E(\phi) = |\phi|^p. \tag{17}$$

The local so-called entropy viscosity is defined for any $K \in \mathcal{T}$ by

$$\mu_K^{\text{Ent}}(\Phi,\mathbf{u}) := C_{\text{Ent}} h_K^2 \frac{\|R^{\text{Ent}}(\Phi,\mathbf{u})\|_{L^\infty(K)}}{\left\|E(\Phi) - \frac{1}{|\Lambda|}\int_\Lambda E(\Phi)\right\|_{L^\infty(\Lambda)}}, \tag{18}$$

where C_{Ent} is an absolute constant. In the regions where ϕ is discontinuous (or has a very sharp gradient), the entropy viscosity as defined earlier may be too large and thereby introduce too much diffusion, which in turn may severely limit the Courant-Friedrichs-Lewy (CFL) number when using an explicit time stepping. In this case, a linear first-order viscosity is turned on instead

$$\mu_K^{\text{Lin}}(\Phi,\mathbf{u}) = C_{\text{Lin}} \left\| h_K \left(\mathbf{u} + \lambda \text{sign}(\Phi)\frac{\nabla\Phi}{\|\nabla\Phi\|_{\ell^2}}\right)\right\|_{L^\infty(K)}, \tag{19}$$

where C_{Lin} is an absolute constant. The justification for the definition of the local speed that is used to define the viscosity in (19) is that (10) can be rewritten $\frac{\partial}{\partial t}\phi + \mathbf{w}\cdot\nabla\phi = \lambda\text{sign}(\phi)\mathcal{G}(\phi)$, where $\mathbf{w} = \mathbf{u} + \lambda\text{sign}(\phi)\frac{\nabla\phi}{\|\nabla\phi\|_{\ell^2}}$. Combining the two viscosities yield the artificial viscosity μ^{Stab} : $\Lambda \times [0,T] \to \mathbb{R}$ defined on each $K \in \mathcal{T}$ by

$$\mu^{\text{Stab}}(\Phi,\mathbf{u})|_K := \min(\mu_K^{\text{Lin}}(\Phi,\mathbf{u}), \mu_K^{\text{Ent}}(\Phi,\mathbf{u})). \tag{20}$$

Going back to the space discretization, we modify (16) as follows: Look for $\Phi \in \mathcal{C}^1([0,T];\mathbb{W}_L(\mathcal{T}))$ so that

$$\int_\Lambda \frac{\partial}{\partial t}\Phi W = -\int_\Lambda \mathbf{u}\cdot\nabla\Phi\, W + \int_\Lambda \lambda\text{sign}(\Phi)\left(\mathcal{G}(\Phi) - \|\nabla\Phi\|_{\ell^2}\right) W - \int_\Lambda \mu^{\text{Stab}}(\Phi,\mathbf{u})\nabla\Phi\cdot\nabla W \tag{21}$$

for all $W \in \mathbb{W}_0(\mathcal{T})$ and $\Phi(\cdot,0) = \Phi_0$.

3.1.3. Approximation in time. Before introducing the time discretization, we rewrite (21) as follows:

$$\int_\Lambda \frac{\partial}{\partial t}\Phi\, W = \int_\Lambda L(\Phi,\mathbf{u})W - \int_\Lambda \mu^{\text{Stab}}(\Phi,\mathbf{u})\nabla\Phi\cdot\nabla W, \qquad \forall W \in \mathbb{W}_0(\mathcal{T},t), \tag{22}$$

where $L(\Phi,\mathbf{u}) := -\mathbf{u} + \lambda\text{sign}(\Phi)\left(\mathcal{G}(\Phi) - \|\nabla\Phi\|_{\ell^2}\right)$. Then we approximate time in the preceding nonlinear system of ODEs by using an explicit RK3 SSP scheme, for example, see [16, 17] for more details on SSP methods. We denote by Φ^k the approximation of $\Phi(\cdot,t^k)$, $0 \leqslant k \leqslant N$. Then, the time stepping proceeds as follows: Given Φ^n, compute $\Phi^1, \Phi^2, \Phi^3 \in W(\mathcal{T})$ and $\Phi^{n+1} \in \mathbb{W}_L(\mathcal{T})$ so that

$$\begin{aligned}
&\text{(i)} && \int_\Lambda \Phi^{(1)}\, W = \int_\Lambda \left(\Phi^n + \delta t^{n+1} L(\Phi^n,\mathbf{u}^n)\right) W, \qquad \forall W \in \mathbb{W}(\mathcal{T}),\\
&\text{(ii)} && \int_\Lambda \Phi^{(2)}\, W = \int_\Lambda \left(\frac{3}{4}\Phi^n + \frac{1}{4}\Phi^{(1)} + \frac{1}{4}\delta t^{n+1} L(\Phi^{(1)},\mathbf{u}^{n+1})\right) W\\
&&& \qquad - \int_\Lambda \mu^{\text{Stab},n+1}\nabla\Phi^{(1)}\cdot\nabla W, \qquad \forall W \in \mathbb{W}(\mathcal{T}),\\
&\text{(iii)} && \int_\Lambda \Phi^{(3)}\, W = \int_\Lambda \left(\frac{1}{3}\Phi^n + \frac{2}{3}\Phi^{(2)} + \frac{2}{3}\delta t^{n+1} L(\Phi^{(2)},\mathbf{u}^{n+\frac{1}{2}})\right) W\\
&&& \qquad - \int_\Lambda \mu^{\text{Stab},n+\frac{1}{2}}\nabla\Phi^{(2)}\cdot\nabla W, \qquad \forall W \in \mathbb{W}(\mathcal{T}),
\end{aligned} \tag{23}$$

and

$$\Phi^{n+1}(\mathbf{x},t) = \begin{cases} \Phi^{(3)}(\mathbf{x}) & \text{when } \mathbf{x} \notin \Gamma_L,\\ upPhi_L(\mathbf{x},t^{n+1}) & \text{when } \mathbf{x} \in \Gamma_L. \end{cases} \tag{24}$$

Using (20), the viscosities $\mu^{\text{Stab},n+1}$ and $\mu^{\text{Stab},n+\frac{1}{2}}$ are defined to be equal to $\mu^{\text{Stab}}(\Phi^{(1)},\mathbf{u}^{n+1})$ and $\mu^{\text{Stab}}\left(\Phi^{(2)},\mathbf{u}^{n+\frac{1}{2}}\right)$, respectively, where the residual R^{Ent} in the definition (18) of μ^{Ent} is evaluated as follows:

$$\begin{aligned}
R(\Phi^{(1)},\mathbf{u}^{n+1}) \approx{}& \frac{E(\Phi^{(1)}) - E(\Phi^n)}{\delta t^{n+1}}\\
&+ \left(\mathbf{u}(t^{n+1})\cdot\nabla\Phi^{(1)} - \lambda\text{sign}(\Phi^{(1)})\left(\mathcal{G}(\Phi^{(1)}) - \|\nabla\Phi^{(1)}\|_{\ell^2}\right)\right) E'(\Phi^{(1)}),
\end{aligned} \tag{25}$$

$$
\begin{aligned}
R(\Phi^{(2)}, \mathbf{u}^{n+\frac{1}{2}}) \approx & \frac{E(\Phi^{(2)}) - E(\Phi^n)}{\frac{1}{2}\delta t^{n+1}} \\
& + \left(\mathbf{u}\left(t^{n+\frac{1}{2}}\right) \cdot \nabla\Phi^{(2)} - \lambda \mathrm{sign}(\Phi^{(2)}) \left(\mathcal{G}(\Phi^{(2)}) - \|\nabla\Phi^{(2)}\|_{\ell^2}\right)\right) E'(\Phi^{(2)}).
\end{aligned} \tag{26}
$$

Remark 1 ('On the Fly' stabilization)
Notice that following [13], no viscosity is added in the computation of $\Phi^{(1)}$. In particular, the viscosities used within the time interval $[t^n, t^{n+1}]$ only depend on the values of Φ and $\mathbf{u}$ on the same time interval.

Remark 2 (Time-step criterion, stability, and convergence)
We expect the scheme to be stable under the following CFL condition:

$$
\delta t^{n+1} \leqslant C_{\mathrm{CFL}} \min_{K \in \mathcal{T}} \frac{h_K}{\|\mathbf{u}\,(t^{n+1}) + \lambda \mathrm{sign}(\Phi^n) \frac{\nabla\Phi^n}{\|\nabla\Phi^n\|_{\ell^2}} \|_{L^\infty(K)}}, \tag{27}
$$

for some sufficiently small but positive constant C_{CFL} independent of $\lambda, \mathcal{T}, \delta t, \Phi$, and $\mathbf{u}$. Refer for instance to [13] for further details on the CFL condition. Moreover, it seems reasonable to expect that only the entropy viscosity is active in the regions where ϕ is smooth; as a result, the CFL condition implies that the first-order approximation of the time derivative in the evaluation of the entropy residuals, (25) and (26), does not affect the overall third-order approximation thanks to the h_K^2 factor present in the definition of the viscosity (18). This conjecture is confirmed numerically in Section 4.2.

3.2. Numerical approximation of the Navier–Stokes system

The space approximation of the velocity and pressure in the Navier–Stokes equations is carried out by using Taylor–Hood finite elements. The time discretization is carried out by using the BDF2. An incremental rotational pressure correction scheme is adopted to uncouple the velocity and the pressure. We refer to [18] for the convergence analysis of the method and to [19] for a review on projection methods.

3.2.1. The space discretization. Let $\mathbf{F}_D$ be a continuous, piecewise quadratic approximation of $\mathbf{f}_D$ on Γ_D. The finite element discretization of the velocity and the pressure is carried out by using the following linear and affine spaces:

$$
\mathbb{V}_0(\mathcal{T}) := \left\{\mathbf{V} \in C^0\left(\overline{\Lambda}; \mathbb{R}^d\right) \mid \mathbf{V}|_{\Gamma_D} = 0,\ \mathbf{V}\cdot\mathbf{u}|_{\Gamma_S} = 0,\ \mathbf{V}|_K \in \mathbb{Q}^2(K),\ \forall K \in \mathcal{T}\right\}, \tag{28}
$$

$$
\mathbb{V}_D(\mathcal{T}) := \left\{\mathbf{V} \in C^0\left(\overline{\Lambda}; \mathbb{R}^d\right) \mid \mathbf{V}|_{\Gamma_D} = \mathbf{F}_D,\ \mathbf{V}\cdot\mathbf{u}|_{\Gamma_S} = 0,\ \mathbf{V}|_K \in \mathbb{Q}^2(K),\ \forall K \in \mathcal{T}\right\}, \tag{29}
$$

$$
\mathbb{M}(\mathcal{T}) := \left\{Q \in C^0\left(\overline{\Lambda}\right) \to \mathbb{R} \mid Q|_K \in \mathbb{Q}^1(K),\ \forall K \in \mathcal{T}\right\}. \tag{30}
$$

Upon setting $\mathbf{U}(0) = \mathbf{U}_0$, where $\mathbf{U}_0$ is a continuous, piecewise quadratic approximation of the initial velocity $\mathbf{u}_0$ in $\mathbb{V}_D(\mathcal{T})$, the semi-discrete formulation of (1) consists of looking for $\mathbf{U} \in \mathcal{C}^1([0,T]; \mathbb{V}_D(\mathcal{T}))$ and $P \in \mathcal{C}^0([0,T]; \mathbb{M}(\mathcal{T}))$ such that the following holds for every $t \in (0,T]$:

$$
\begin{aligned}
&\int_\Lambda \rho\left(\frac{\partial}{\partial t}\mathbf{U} + (\mathbf{U}\cdot\nabla)\mathbf{U}\right)\cdot\mathbf{V} + 2\int_\Lambda \mu\left(\nabla^S\mathbf{U} : \nabla^S\mathbf{V}\right) - \int_\Lambda P\,\mathrm{div}\mathbf{V} + \int_\Lambda Q\,\mathrm{div}\mathbf{U} \\
&= \int_\Lambda \rho\mathbf{g}\cdot\mathbf{V} + \int_{\Gamma_N} \mathbf{f}_N\cdot\mathbf{V} + \int_{\Sigma(t)} \sigma\kappa\,\mathbf{n}\cdot\mathbf{V}, \qquad \forall(\mathbf{V}, Q) \in \mathbb{V}_0(\mathcal{T}) \times \mathbb{M}(\mathcal{T}).
\end{aligned} \tag{31}
$$

3.2.2. Time discretization. The time discretization of (31) is performed by using the BDF2 and an incremental pressure correction scheme in rotational form introduced and studied in [20] to decouple the velocity and the pressure. In addition to the approximation of the initial condition

$\mathbf{U}^0$ mentioned earlier, the algorithm requires an approximation $P_0 \in \mathbb{M}(\mathcal{T})$ of the initial pressure $p(0)$. We denote by $\mathbf{U}^n$, P^n, and Ψ^n the approximations of $\mathbf{U}(\cdot, t^n)$ and $P(\cdot, t^n)$ and the pressure correction, respectively.

The initialization step consists of setting: $\mathbf{U}^0 = \mathbf{U}_0, P^{-1} = P^0 = P_0$ and $\Psi^0 = 0$. Then, given a new time step δt^{n+1}, given $\mathbf{U}^n \in \mathbb{V}_D(\mathcal{T}), \Psi^n \in \mathbb{M}(\mathcal{T})$, and $P^n \in \mathbb{M}(\mathcal{T})$ and assuming for the time being that $\rho(t^{n+1}), \mu(t^{n+1})$, and $\Sigma(t^{n+1})$ are also known (Section 3.4), the new fields $\mathbf{U}^{n+1} \in \mathbb{V}_D(\mathcal{T}), \Psi^{n+1} \in \mathbb{M}(\mathcal{T})$ and $P^{n+1} \in \mathbb{M}(\mathcal{T})$ are computed in three steps:

$\boxed{1}$*Velocity prediction:* Find $\mathbf{U}^{n+1} \in \mathbb{V}_D(\mathcal{T})$ such that

$$\begin{aligned}
&\int_\Lambda \rho\left(t^{n+1}\right) \frac{\delta_{\mathrm{BDF2}}}{\delta t^{n+1}} \mathbf{U}^{n+1} \cdot \mathbf{V} + 2 \int_\Lambda \mu\left(t^{n+1}\right)\left(\nabla^S\left(\mathbf{U}^{n+1}\right) : \nabla^S \mathbf{V}\right) + S_{\mathcal{T}}\left(\mathbf{U}^{n+1}, \mathbf{V}\right) \\
&= -\int_\Lambda \rho\left(t^{n+1}\right)\left(\left(\mathbf{U}^n\right)^\star \cdot \nabla \mathbf{U}^n\right) \cdot \mathbf{V} + \int_\Lambda \left(P^n + \frac{4}{3}\Psi^n - \frac{1}{3}\Psi^{n-1}\right) \operatorname{div}(\mathbf{V}) \\
&\quad + \int_\Lambda \rho^{n+1} \mathbf{g}^{n+1} + \int_{\Gamma_N} \mathbf{f}_N^{n+1} \cdot \mathbf{V} + \int_{\Sigma(t^{n+1})} \sigma \kappa^{n+1}\, \mathbf{n}^{n+1} \cdot \mathbf{V}, \quad \forall \mathbf{V} \in \mathbb{V}_0(\mathcal{T}),
\end{aligned} \tag{32}$$

where

- the BDF2 approximation of the time derivative with variable time stepping is given by

$$\begin{aligned}
\frac{\partial}{\partial t} \mathbf{U}\left(\mathbf{x}, t^{n+1}\right) &\approx \frac{\delta_{\mathrm{BDF2}}}{\delta t^{n+1}} \mathbf{U}^{n+1}(\mathbf{x}) \\
&:= \frac{1}{\delta t^{n+1}} \left(\frac{1 + 2\eta_{n+1}}{1 + \eta_{n+1}} \mathbf{U}^{n+1}(\mathbf{x}) - \left(1 + \eta_{n+1}\right) \mathbf{U}^n(\mathbf{x}) + \frac{\eta_{n+1}^2}{1 + \eta_{n+1}} \mathbf{U}^{n-1}(\mathbf{x}) \right),
\end{aligned}$$

  where $\eta_{l+1} := \frac{\delta t^{l+1}}{\delta t^l}$.
- the extrapolated velocity $(\mathbf{U}^n)^\star$ is defined by $(\mathbf{U}^n)^\star := \mathbf{U}^n + \eta^n(\mathbf{U}^n - \mathbf{U}^{n-1})$;
- as discussed in [21], the bilinear form $S_{\mathcal{T}} : \mathbb{V}(\mathcal{T}) \times \mathbb{V}(\mathcal{T}) \to \mathbb{R}$ is added to control the divergence of the velocity and to cope with variable time stepping and open boundary conditions

$$S_{\mathcal{T}}(\mathbf{W}, \mathbf{V}) := C_{\mathrm{stab}} \sum_{K \in \mathcal{T}} \int_K \left(\mu^{n+1} + \rho^{n+1} \|h_K \mathbf{W}\|_{L^\infty(K)}\right) (\nabla \cdot \mathbf{W})(\nabla \cdot \mathbf{V}), \tag{33}$$

  where C_{stab} is an absolute constant.

$\boxed{2}$*Pressure-correction step:* The pressure increment $\Psi^{n+1} \in \mathbb{M}(\mathcal{T})$ is determined by solving

$$\int_\Lambda \nabla \Psi^{n+1} \nabla Q = -\frac{3 \min_{x \in \Lambda} \rho\left(t^{n+1}\right)}{2 \delta t^{n+1}} \int_\Lambda \operatorname{div}\left(\mathbf{U}^{n+1}\right) Q, \ \ \forall Q \in \mathbb{M}(\mathcal{T}). \tag{34}$$

$\boxed{3}$*Pressure update:* The pressure $P^{n+1} \in \mathbb{M}(\mathcal{T})$ is obtained by solving

$$\int_\Lambda P^{n+1} Q = \int_\Lambda \left(P^n + \Psi^{n+1}\right) Q - \min_{x \in \Lambda} \mu\left(t^{n+1}\right) \int_\Lambda \operatorname{div0}\left(\mathbf{U}^{n+1}\right) Q, \ \forall Q \in \mathbb{M}(\mathcal{T}). \tag{35}$$

For stability purposes, we restrict the space and time discretization parameters to satisfy a CFL condition

$$\delta t^{n+1} \leqslant C_{\mathrm{CFL}} \frac{1}{2} \frac{\min_{K \in \mathcal{T}_h} h_K}{\|\mathbf{U}^n\|_{L^\infty(\Lambda)}}, \tag{36}$$

where C_{CFL} is the same constant appearing in (27) and $\frac{1}{2} \min_{K \in \mathcal{T}_h} h_K$ is the minimum distance between two Lagrange nodes using $\mathbb{Q}_2$ elements.

DOI: 10.1002/fld

3.3. The surface tension

This section describes the approximation of the curvature term $\int_{\Sigma(t^{n+1})} \sigma\kappa^{n+1}\,(\mathbf{n}^{n+1}\cdot\mathbf{V})$ appearing in the first step of the projection method (32). We follow the method proposed in [22] (see also [23]) using the work in [24]. This approach is based on the following representation of total curvature:

$$\kappa\mathbf{n} = \nabla_\Sigma\cdot\nabla_\Sigma\mathbf{Id}_\Sigma,$$

where $\mathbf{Id}_\Sigma$ is the identity mapping on Σ and, given any extension $\tilde{\mathbf{v}}$ of $\mathbf{v}$ in a neighborhood of Σ, the tangential gradient of $\mathbf{v} : \Sigma \to \mathbb{R}^d$ is defined by $\nabla_\Sigma\mathbf{v} := \nabla\tilde{\mathbf{v}}|_\Sigma(\mathbf{I}-\mathbf{n}\otimes\mathbf{n})$, see, for example, [25]. Multiplying the preceding identity by a test function $\mathbf{V}$ and integrating by parts over Σ yields

$$\int_\Sigma \sigma\kappa\mathbf{n}\cdot\mathbf{V} = -\int_\Sigma \sigma\nabla_\Sigma\mathbf{Id}_\Sigma : \nabla_\Sigma\mathbf{V} + \int_{\partial\Sigma}\sigma\partial_\tau\mathbf{Id}_\Sigma\cdot\mathbf{V}, \tag{37}$$

where τ is the co-normal to Σ and ∂_τ is the derivative in the co-normal direction. In the present context, the integral on $\partial\Sigma$ vanishes because either Σ is a closed manifold or $\partial_\tau\mathbf{Id}_\Sigma = 0$. This identity together with the first-order prediction $\mathbf{Id}_{\Sigma(t^{n+1})} \approx \mathbf{Id}_{\Sigma(t^n)} + \delta t^{n+1}\mathbf{U}^{n+1}$ of the interface evolution (5) gives a semi-implicit representation of the total curvature

$$\begin{aligned}\int_{\Sigma(t^{n+1})}\sigma\kappa^{n+1}\,(\mathbf{n}^{n+1}\cdot\mathbf{V}) &= -\int_{\Sigma(t^{n+1})}\nabla_{\Sigma(t^{n+1})}\mathbf{Id}_{\Sigma(t^{n+1})} : \nabla_{\Sigma(t^{n+1})}\mathbf{V}\\ &\approx -\int_{\Sigma(t^{n+1})}\nabla_{\Sigma(t^{n+1})}\mathbf{Id}_{\Sigma(t^{n+1})} : \nabla_{\Sigma(t^{n+1})}\mathbf{V}\\ &\quad -\delta t^{n+1}\int_{\Sigma(t^{n+1})}\sigma\nabla_{\Sigma(t^{n+1})}\mathbf{U}^{n+1} : \nabla_{\Sigma(t^{n+1})}\mathbf{V}.\end{aligned}$$

One key benefit of this representation comes from the additional stabilizing term $\int_{\Sigma(t^{n+1})}$ $\sigma\nabla_{\Sigma(t^{n+1})}\mathbf{U}^{n+1} : \nabla_{\Sigma(t^{n+1})}\mathbf{V}$, which we keep implicit in (32). The technicalities regarding the approximation of this integral using the level set representation are detailed in Section 3.4.2 (45).

3.4. The coupled system

The two solvers described in Sections 3.1 and 3.2 earlier are sequentially combined. The flowchart of the resulting free boundary flow solver is shown in Figure 3. The remaining subsections of Section 3.4 detail the coupling between the two solvers and other implementation technicalities.

3.4.1. Data for the level set solver (23)–(24). Following (23), given Φ^n, an approximation of $\phi(t^n)$, the approximation Φ^{n+1} of $\phi(t^{n+1})$ requires the velocities $\mathbf{u}\,(t^n)$, $\mathbf{u}\left(t^{n+\frac{1}{2}}\right)$, and $\mathbf{u}\left(t^{n+1}\right)$. To avoid an implicit coupling between the level set solver and the Navier-Stokes solver, these quantities are replaced by second-order extrapolations using $\mathbf{U}^n$ and $\mathbf{U}^{n-1}$:

$$\mathbf{u}(t^n) \approx \mathbf{U}^n,\ \mathbf{u}\left(t^{n+\frac{1}{2}}\right) \approx \mathbf{U}^n + \frac{1}{2}\frac{\delta t^{n+1}}{\delta t^n}\left(\mathbf{U}^n - \mathbf{U}^{n-1}\right),\ \text{and}\ \mathbf{U}^{n+1} \approx \mathbf{U}^n + \frac{\delta t^{n+1}}{\delta t^n}\left(\mathbf{U}^n - \mathbf{U}^{n-1}\right).$$

The sign(.) function, which is used in the right-hand side of (10) to make sure that $\|\nabla\phi\|_{\ell^2}$ is close to 1 in a small neighborhood of Σ (i.e., on the fly reinitialization, see discussion following (11)), is redefined and replaced by:

$$\text{sign}_h(s) = \begin{cases} +1 & \text{if } s > \beta\tanh(C_S),\\ -1 & \text{if } s < -\beta\tanh(C_S),\\ 0 & \text{otherwise,}\end{cases} \tag{38}$$

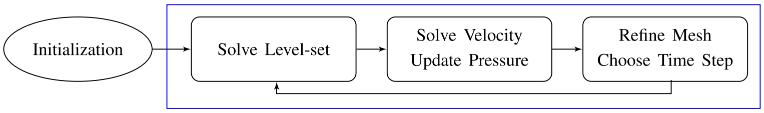


Figure 3. Global algorithm flowchart.

where C_S is an absolute constant. The thresholding in the preceding definition of the approximate sign function gives $\text{sign}_h(\phi(\mathbf{x},t)) = \pm 1$ whenever $\frac{|\phi(\mathbf{x},t)|}{\beta} \geqslant \tanh(C_S)$, which is compatible with the behavior $\frac{\phi(\mathbf{x},t)}{\beta} \approx \tanh\left(\frac{\text{dist}(\Sigma(t),\mathbf{x})}{\beta}\right)$ that is expected for the level set function (12).

The parameter λ ('reinitialization relative speed" in the language of [14]) is defined for $t \in (t^n, t^{n+1}]$ by

$$\lambda = C_\lambda \|\mathbf{U}^n\|_{L^\infty(\Lambda)}, \qquad K \in \mathcal{T}, \tag{39}$$

where C_λ is an absolute constant. This definition is motivated by the CFL condition (27). We refer to [14] for further details.

3.4.2. Data for the Navier–Stokes solver. The definitions of the fields $\mathbf{U}^{n+1}$ and P^{n+1} in (32)–(35) invoke the values of the density field ρ^{n+1} and viscosity field μ^{n+1}. Once ϕ^{n+1} is computed, these quantities are evaluated by using the following definitions

$$\rho^{n+1} = \rho^+ \frac{1 + H_h(\Phi^{n+1})}{2} + \rho^- \frac{1 - H_h(\Phi^{n+1})}{2}, \tag{40}$$

$$\mu^{n+1} = \mu^+ \frac{1 + H_h(\Phi^{n+1})}{2} + \mu^- \frac{1 - H_h(\Phi^{n+1})}{2}, \tag{41}$$

where $\rho^\pm, \mu^\pm$ are the density/viscosity in $\Omega^\pm$, and $H_h(.)$ an approximation of the Heaviside function defined as follows:

$$H_h(s) = \begin{cases} 1 & \text{if } s > \beta \tanh(C_H), \\ -1 & \text{if } s < -\beta \tanh(C_H), \\ \dfrac{s}{\beta \tanh(C_H)} & \text{otherwise,} \end{cases} \tag{42}$$

where C_H is an absolute constant, as suggested in [14]. Similarly to what we have carried out to approximate the sign function, the preceding regularization is compatible with the behavior $\frac{\phi(\mathbf{x},t)}{\beta} \approx \tanh\left(\frac{\text{dist}(\Sigma(t),\mathbf{x})}{\beta}\right)$ that is expected for the level set function.

The approximation of the surface tension term in (32) is performed by following [23]. Let $\epsilon > 0$, and consider the piecewise linear regularized Dirac measure supported on $\Sigma(t^{n+1})$, say δ_ϵ, defined by

$$\delta_\epsilon(\mathbf{x}) := \begin{cases} \dfrac{1}{\epsilon}\left(1 - \frac{\text{dist}(\Sigma,\mathbf{x})}{\epsilon}\right) & \text{if } |\text{dist}(\Sigma,\mathbf{x})| < \epsilon, \\ 0 & \text{otherwise.} \end{cases} \tag{43}$$

To account for the fact that we do not have access to the distance to the interface Σ but rather to an approximation of $\beta \tanh\left(\frac{\text{dist}(\Sigma,\mathbf{x})}{\beta}\right)$ (Section 2.2.2), we rescale $\delta_\epsilon(\mathbf{x})$ as suggested in [26, 27] and consider instead

$$\delta_\epsilon(\mathbf{x},\Phi) = \begin{cases} \dfrac{1}{\tilde{\epsilon}}\left(1 - \frac{\Phi(\mathbf{x})}{\tilde{\epsilon}}\right) \|\nabla\Phi(\mathbf{x})\|_{\ell^2} & \text{if } |\Phi(\mathbf{x})| < \tilde{\epsilon} := \epsilon \frac{\|\nabla\Phi(\mathbf{x})\|_{\ell^1}}{\|\nabla\Phi(\mathbf{x})\|_{\ell^2}}, \\ 0 & \text{otherwise,} \end{cases} \tag{44}$$

where $\|\cdot\|_{\ell^p}$ is the ℓ^p-norm in $\mathbb{R}^d$. In practice, we chose $\epsilon = \beta \tanh(C_H)$ to be consistent with the approximation of the Heaviside function (42). Using this approximate Dirac measure, the approximation of the surface tension discussed in Section 3.3 becomes

$$\begin{aligned}\int_{\Sigma(t^{n+1})} \sigma\kappa^{n+1}\left(\mathbf{n}^{n+1}.\mathbf{V}\right) \approx & -\int_{\Sigma(t^{n+1})} \nabla_{\Sigma(t^{n+1})}\mathbf{Id}_{\Sigma(t^{n+1})} : \nabla_{\Sigma(t^{n+1})}\mathbf{V} \\ & - \delta t^{n+1}\int_{\Sigma(t^{n+1})} \sigma\nabla_{\Sigma(t^{n+1})}\mathbf{U}^{n+1} : \nabla_{\Sigma(t^{n+1})}\mathbf{V} \\ \approx & -\int_{\Lambda} \left(\nabla_{\Phi^{n+1}}\mathbf{Id}_{\Lambda} : \nabla_{\Phi^{n+1}}\mathbf{V}\right)\delta_{\epsilon}(.,\Phi^{n+1}) \\ & - \delta t^{n+1}\int_{\Lambda}\left(\sigma\nabla_{\Phi^{n+1}}\mathbf{U}^{n+1} : \nabla_{\Phi^{n+1}}\mathbf{V}\right)\delta_{\epsilon}(.,\Phi^{n+1}),\end{aligned}$$

where $\mathbf{Id}_\Lambda$ is the identity mapping on Λ and

$$\begin{aligned}\nabla_{\Phi^{n+1}}\mathbf{Id}_{\Lambda} &= I - \frac{\nabla\Phi^{n+1}\otimes\nabla\Phi^{n+1}}{\|\nabla\Phi^{n+1}\|_{\ell^2}^2}, \\ \nabla_{\Phi^{n+1}}\mathbf{V} &:= \nabla\mathbf{V}\left(I - \frac{\nabla\Phi^{n+1}\otimes\nabla\Phi^{n+1}}{\|\nabla\Phi^{n+1}\|_{\ell^2}^2}\right), \quad \forall\mathbf{V}\in\mathbb{V}_D(\mathcal{T}).\end{aligned}$$

Note that the preceding definitions correspond to approximating the normal vector $\mathbf{n}$ on $\Sigma(t^{n+1})$ by $-\frac{\nabla\Phi^{n+1}}{\|\nabla\Phi^{n+1}\|_{\ell^2}}$.

3.5. Adaptive mesh refinement

The experiments reported in [1, 6, 7] suggest that a critical feature of bouncing jets is the occurrence of a thin layer between the jet and the rest of the fluid. These observations have led us to adopt a mesh-refinement technique to describe accurately this thin layer. We propose here a refinement strategy solely based on the position of the interface by increasing the mesh resolution around the zero level set of Φ. More precisely, a cell K is refined if its generation count (the number of times a cell from the initial subdivision has been refined to produce the current cell) is smaller than a given number $R_{\max}$ and if

$$|\Phi(\mathbf{x}_K,t)| \leqslant \beta\tanh(C_R), \tag{45}$$

where $\mathbf{x}_K$ is the barycenter of K and C_R is an absolute constant. The purpose of the parameter $R_{\max}$ is to control the total number of cells. This refinement strategy leads to subdivisions where all the elements crossing the zero level set of Φ have the same diameter (up to a constant only depending on the coarse mesh). The parameter C_R controls the distance to the interface below which refinement occurs. In fact, recalling that using the filter (12) implies that $\phi(\mathbf{x},t) \approx \beta\tanh\left(\frac{\text{dist}(\Sigma(t),\mathbf{x})}{\beta}\right)$ if λ is large enough, we expect the refinement criteria (45) to be active whenever $\text{dist}(\Sigma(t),\mathbf{x}) \leqslant C_R\beta$. Note that (45) is compatible with (38) and (42). The subdivisions are carried out with at most one hanging node per face, see, for example, Bonito and Nochetto [28, Section 6.3]. A cell is coarsen if it satisfies the following three conditions: its generation count is positive;

$$|\Phi(\mathbf{x}_K,t)| \geqslant \beta\tanh(C_C), \tag{46}$$

where $C_C \geqslant C_R$ is an absolute constant; and if once coarsened, the resulting subdivision does not have more than one hanging node per face. We refer to the documentation of the deal.II library for further details [29].

4. NUMERICAL VALIDATIONS

The algorithm presented in the previous sections has been implemented using the *deal.II* finite element library described in [29, 30]. Parallelism is handled by using the message passing interface library [31]. The subdivision and mesh distribution is carried out by using the p4est library from [32].

DOI: 10.1002/fld

The rest of this section illustrates and evaluates the performance of the preceding algorithm. We start in Section 4.1 by specifying all the numerical constants required by the algorithm. The validation of the transport code for solving the level set equation is carried out in Section 4.2. The validation of the two-phase fluid system is performed in Section 4.3.

4.1. Numerical parameters

Our algorithm involves several numerical parameters. In this section, we briefly recall their meaning, where they appear, and we specify the value of each of them. Unless specified otherwise, these values are fixed for this entire section. Except for Section 4.2, the time discretization parameter is always chosen according to (27).

A complete list of all the parameters is shown in Table I. The parameter β determining the width of the hyperbolic tangent filter (12) is defined to be $\beta = \min_{K \in \mathcal{T}} h_K$. This definition is used in the expression of the hyperbolic tangent filter (12), in the threshold of the approximate Heaviside function (42) and the approximate sign function (38), and in the refinement strategy, (45) and (46). In particular, the expected number of elements in the transition layer separating the two phases is $2C_R = 4$ (Section 3.5).

4.2. Transport of the level set

4.2.1. Convergence tests. The consistency of the algorithm for the approximation of the level set Equations (23) and (24) is a priori third-order in time and space. To evaluate whether this is indeed the case, we solve the linear transport equation in the unit square $\Lambda = (0,1)^2$ using the velocity field

$$\mathbf{u}(x,y,t) := \begin{pmatrix} -\sin^2(\pi x)\sin(2\pi y)\cos(\pi t/0.1) \\ \sin^2(\pi y)\sin(2\pi x)\cos(\pi t/0.1) \end{pmatrix}.$$

This flow is time periodic of period 1, which implies that $\phi(\mathbf{x}, 0.2) = \phi_0(\mathbf{x})$, for all $\mathbf{x} \in \Lambda$. The initial level set function is chosen to be the signed distance to the line of equation $y = 0.5$:

$$\phi_0(x,y) = y - 0.5. \tag{47}$$

The errors are evaluated at $t = 0.2$. Three different scenarios are considered: (i) no stabilization and no reinitialization; (ii) entropy viscosity stabilization and no reinitialization; (iii) entropy viscosity and reinitialization, that is, the complete algorithm. Except for β, the values of the numerical parameters are given in Table I. We consider four computations carried out on four uniform meshes

Table I. Description and values of the numerical parameters.

	C_{CFL}	C_λ	C_H	C_S	C_R	C_C	$R_{\max}$	C_{Lin}	C_{Ent}	p	C_{stab}
Purpose	CFL	Reinitialization			Adaptivity			Stabilization			
Appears in	(27) & (36)	(39)	(42)	(38)	(45)	(46)	§3.5	(19)	(18)	(17)	(33)
Value	0.25	0.01	1.25	0.5	2.	2.	2.	0.1	0.1	20	0.1

Table II. Single vortex test case: L^2-norm of error and observed convergence rates for each scenario after one period for different time discretization resolutions.

	Scenario (i)		Scenario (ii)		Scenario (iii)	
δt	L^2 error	Observed rate	L^2 error	Observed rate	L^2 error	Observed rate
1e−2	1.2e−5	—	1.5e−5	—	1.3e−3	—
5e−3	1.5e−6	3	1.8e−6	3.0	1.5e−4	3
2.5e−3	1.9e−7	3	2.0e−7	3.2	2.0e−5	3
1.25e−3	2.4e−8	3	2.5e−8	3.1	2.6e−6	3

(i) Runge–Kutta 3 (RK3) Galerkin algorithm; (ii) RK3 algorithm with entropy viscosity; (iii) RK3 with entropy viscosity and reinitialization. Third-order convergence rate is observed in all scenarios.

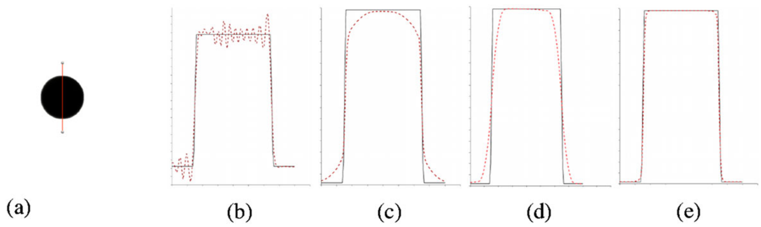


Figure 4. (a) Initial level set: the dark region corresponds to $\phi_0 \geqslant 0$; (b)–(e) comparison of the graphs of the exact (solid line) and approximate (dashed line) level set functions along the line of equation $x = 0.5$ after one revolution: (b) without stabilization and reinitialization ($C_{\text{Lin}} = C_{\text{Ent}} = C_\lambda = 0$); (c) with first-order linear stabilization, $C_{\text{Lin}} = 0.1$; (d) with entropy residual stabilization, $C_{\text{Lin}} = 0.1$ and $C_{\text{Ent}} = 0.1$; and (e) with entropy residual stabilization and reinitialization, $C_\lambda = 0.01$. The transport without stabilization of the nearly discontinuous level set function yields spurious oscillations. These oscillations are removed by the linear viscosity at the expense of large numerical diffusion. The numerical diffusion is minimized by turning on the entropy residual term. Finally, adding the reinitialization allows to nearly recover the exact profile.

with constant time steps. The mesh size and the time step are divided by 2 each time. The meshes are composed of 1089, 4225, 16,64, and 66,049 $\mathbb{Q}_2$ degree of freedoms. The space discretization is chosen fine enough not to influence the time error. We report in Table II, the errors $\|\Phi^N - \phi_0\|_{L^2(\Lambda)}$ for the three scenarios and the observed rates of convergence. Note that the reinitialization is turned on in scenario (iii), which implies that the exact solution at $t = 0.2$ is not ϕ_0 anymore; we must instead compare Φ^N with $\beta \tanh\left(\frac{\phi_0(x,y)}{\beta}\right)$. In this particular case, we keep β constant to ascertain the third-order consistency of the algorithm; we set $\beta = 0.0203125$.

4.2.2. Rotation of a circular level set: Effect of different viscosities and reinitialization. In this test case, the initial data for the level set are the hyperbolic tangent filter applied to the distance to the circle centered at (0.5,0) and of radius 0.25

$$\phi_0(x,y) := \beta \tanh\left(\frac{0.25 - ((x-0.5)^2 + y^2)^{\frac{1}{2}}}{\beta}\right), \qquad (x,y) \in \Lambda := (-1,1)^2,$$

see Figure 4(a). The level set is transported by using the solid rotation velocity field:

$$\mathbf{u}(x,y) := \begin{pmatrix} -y \\ x \end{pmatrix}. \tag{48}$$

The computation is carried out with the adaptive algorithm described in Section 3.5 using $R_{\max} = 2$. The initial mesh is uniform with $h_K = 0.015625$ for all $K \in \mathcal{T}$. The resulting time-dependent mesh is such that $\min_{T \in \mathcal{T}} h_K \approx 0.00390625$, that is, we set $\beta = 0.00390625$. The values of the other numerical constants are provided in Table I. The time step δt is chosen to be uniform and equal to $\pi \times 10^{-3}$. We compare the initial level set with its approximation after one revolution. Figure 4(b)–(e) illustrates the benefits of using the entropy stabilization and the reinitialization technique by comparing the graphs of the exact and approximate level set functions along the line of equation $x = 0.5$ after one revolution.

4.2.3. 3D slotted disk: long-time behavior. A typical benchmark for the transport of a level set function is the so-called Zalesak disk documented in [33]. We consider in this section the three-dimensional version thereof. The computational domain is $\Lambda := (0,1)^3$. The initial level set is the characteristic function of a slotted sphere centered at (0.5,0.75,0.5) with a radius of 0.15. The width, height, and depth of the slot are 0.0375, 0.15, and 0.3 respectively (Figure 5(a)). The initial profile is transported by using the following velocity field:

$$\mathbf{u}(x,y,z) := \begin{pmatrix} 0.5 - y \\ x - 0.5 \\ 0 \end{pmatrix}. \tag{49}$$

DOI: 10.1002/fld

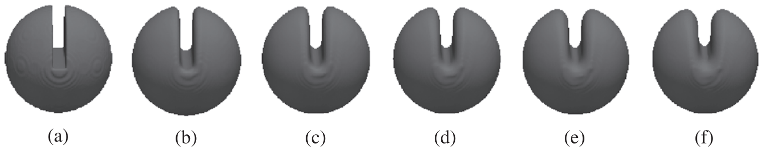


Figure 5. Rotating slotted three-dimensional sphere. From left to right, the dark regions correspond to $\Phi \geqslant 0$ after 0,1,2,3,4, and 5 full rotations. Oscillations and numerical diffusion are controlled by the entropy viscosity as well as the reinitialization algorithm.

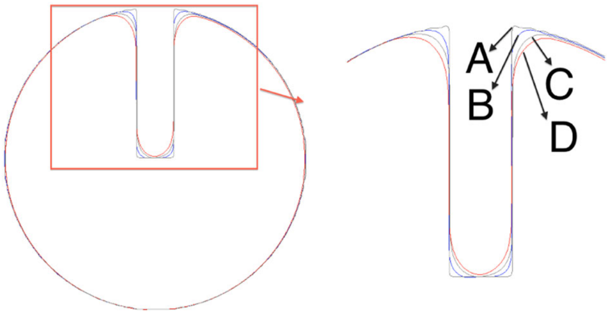


Figure 6. Iso-contour $\Phi = 0$ in the plane $z = 0$: (A) initial data; (B) after one rotation; (C) after three rotations; and (D) after five rotations. Numerical diffusion is observed but is greatly minimized by the entropy viscosity and reinitialization techniques.

The time step is chosen to be uniform $\delta t = \pi \times 10^{-3}$. The initial subdivision is composed of cells of diameter 0.015625. The adaptive mesh refinement technique described in Section 3.5 is used with $R_{\max} = 2$; the minimum mesh size is $\min_{K \in \mathcal{T}} h_K = 0.00390625$. The numerical constants are given in Table I. The computation is performed until the slotted sphere has undergone five full revolutions. The iso-surface $\Phi = 0$ is shown in Figure 5 after each of the five periods. Oscillations and numerical diffusion are controlled by the entropy viscosity and the reinitialization algorithm. A closer look at the slotted region is provided in Figure 6.

4.2.4. Single vortex: large deformations. The single vortex problem consists of the deformation of a sphere by a time-periodic incompressible vortex-like flow. The computational domain is $\Lambda = (0,1)^3$, and the time-periodic velocity field is defined by

$$\mathbf{u}(x,y,t) := \begin{pmatrix} -\sin^2(\pi x)\sin(2\pi y)\cos(\pi t/4) \\ \sin^2(\pi y)\sin(2\pi x)\cos(\pi t/4) \\ 0 \end{pmatrix}. \tag{50}$$

The initial level set is given by

$$\phi_0(\mathbf{x}) := \beta \tanh\left(\frac{\operatorname{dist}(S,\mathbf{x})}{\beta}\right), \tag{51}$$

where S is the sphere centered at (0.5,0.75,0.5) of radius of 0.15. The field ϕ_0 is a regularized version of the distance function using the tanh cut-off filter (12). The divergence-free velocity field severely deforms the level set until $t = 2$ and returns it to its initial shape at $t = 4$. The time step is chosen to be uniform $\delta t = 0.001$ and the final time is $t = 4$ (one cycle). The initial subdivision is made of uniform cells of diameter 0.015625; the minimum mesh size allowed in the adaptive mesh refinement is 0.00390625. The numerical constants are given in Table I. Figure 7 shows the iso-contour $\Phi = 0$ at different times. The undeformed sphere is recovered after one cycle.

4.3. Two-phase flows

4.3.1. Rising bubble: surface tension benchmark. We start the validation of the two-phase flow system with the Rising bubble benchmark problem, see, for example, [23]. The computational domain is $\Lambda = (0,1) \times (0,2)$, the initial data ϕ_0 are the characteristic function of a circular bubble of radius

DOI: 10.1002/fld

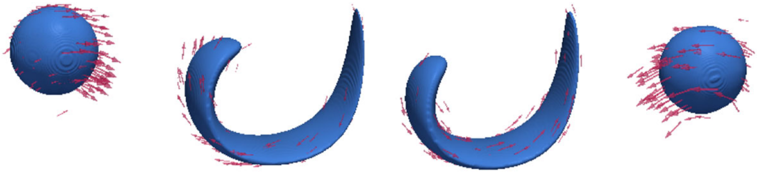


Figure 7. 3D Vortex using (50). From left to right: $t = 0, t = 1, t = 3$, and $\mathrm{t} = 4$. The initial sphere is recovered after one cycle. The vectors indicate the direction of the velocity field.

Table III. Two different sets of physical constants for the rising bubble benchmark problems.

Test case	ρ^+	ρ^-	μ^+	μ^-	g	σ
1	1000	100	10	1.0	0.98	24.5
2	1000	1	10	0.1	0.98	1.96

The first test case consists of a density/viscosity ratio of 10 with a large surface tension coefficient. The density/viscosity ratio is equal to 100 in the second case and the surface tension coefficient is smaller.

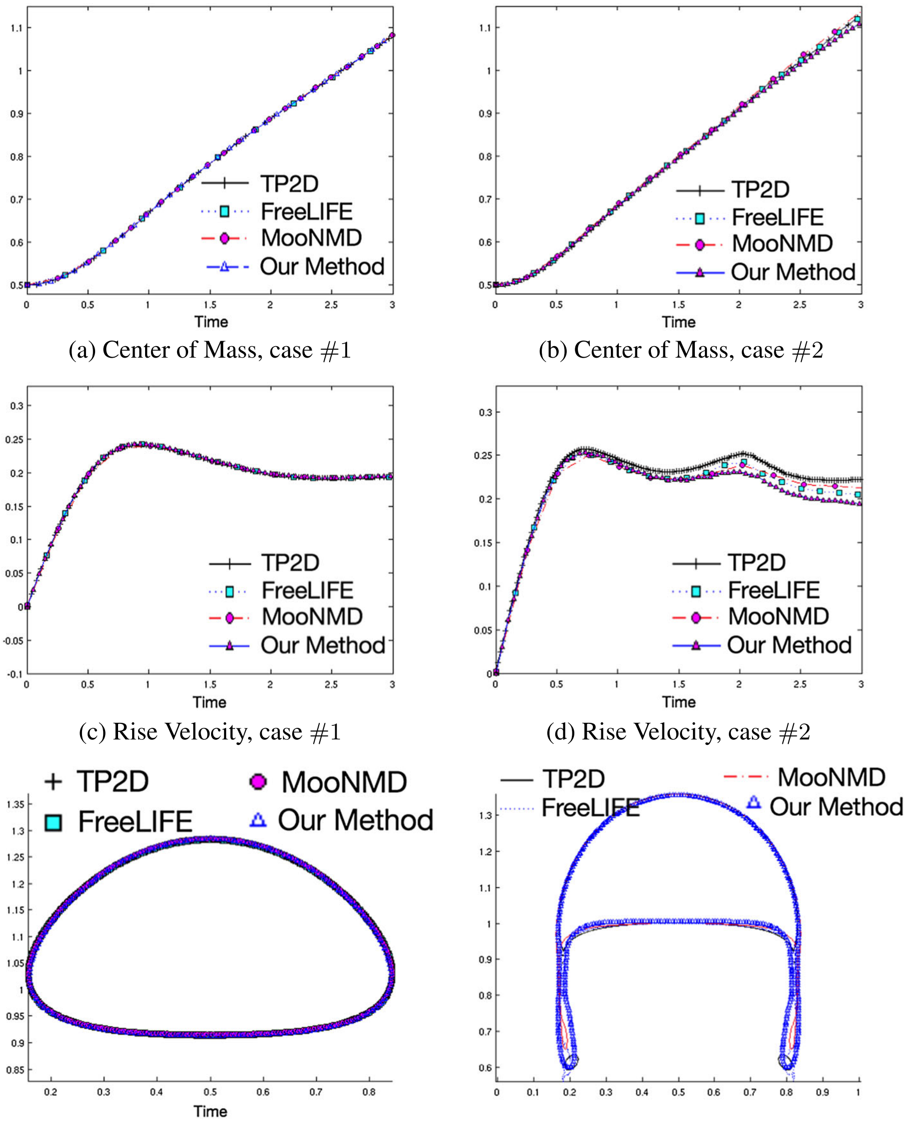
0.25 centered at $(0.5, 0.5)$. Two different sets of physical constants are considered (Table III); σ is the surface tension coefficient and g is the magnitude of the non-dimensional gravitational force. The domains Ω^+ and Ω^- are the domains inside and outside the bubble, respectively. The no-slip boundary condition is imposed at the top and bottom of the computational domain. The free-slip condition (2c) is imposed on the side walls. The initial subdivision is made of uniform cells of diameter 0.03125; the minimum mesh size allowed in the adaptive mesh refinement is 0.0078125. The time steps δt are chosen uniform according to the CFL restriction (27), and the values of the numerical constants are given in Table I. We compare in Figure 8 our results with those from three other methods. We show in panel (a) the time history of the center of mass $\mathbf{X}_c := \int_{\Omega_2} \mathbf{x} d\mathbf{x} / \int_{\Omega_2} 1 d\mathbf{x}$, in panel (b) the rising velocity $u_c := \int_{\Omega_2} u_y d\mathbf{x} / \int_{\Omega_2} 1 d\mathbf{x}$, and in panel (c) the shape of the bubble at $t = 3$. The results of our simulations are within the range of those given by the benchmark algorithms. The shape of the bubble for test case #2 at $t = 3$ is reported in Figure 9 and compared with the shapes obtained by the other algorithms described in [23].

To illustrate the CPU performance of the algorithm, we now report some characteristics times recorded when solving test case #1 using an *Intel(R) Xeon(R) CPU E5-2609 v3 @ 1.90 GHz* processor. The average wall clock time for each time step is approximately 0.3542 s and the computation requires 1536 time steps. The space discretization is adaptively modified after each time step, but overall, there are approximately 9000 degrees of freedom for the velocity approximation and 1000 degrees of freedom for each the pressure and the level set approximations. The average wall clock time per time step and per degree of freedom is approximately 3.5×10^{-5} s.

4.3.2. Buckling fluids. We now test the algorithm in the context of fluid buckling, see, for instance, [14, 35, 36]. The test consists of letting a free-falling jet of very viscous fluid impinge on a horizontal surface. The diameter of the impinging jet is 0.1, and the inflow velocity is $1\mathrm{ms}^{-1}$ in $\Lambda = (0, 1\mathrm{m})^2$. The physical parameters chosen for the falling fluid are $\rho^+ = 1800\,\mathrm{kg\,m}^{-3}, \mu^+ = 250\,\mathrm{Pa\,s}$, and $\rho^- = 1\,\mathrm{kg\,m}^{-3}, \mu^- = 2 \times 10^{-5}$ for the ambient fluid as in [14]. The no-slip boundary condition ($\mathbf{u} = \mathbf{0}$) is imposed at the bottom boundary, and an inflow boundary condition is imposed at the top boundary where $|x - 0.5| < 0.05$. The open boundary condition is enforced on all the other boundaries, that is, $(2\mu\nabla^S\mathbf{u} - pI)\mathbf{u} = 0$. Surface tension is neglected for this test case ($\sigma = 0$). The numerical constants are given in Table I.

The initial subdivision is composed of cells of uniform diameter 0.015625 m, and the minimum cell diameter reached during the mesh adaption process is 0.003906 m. The time evolution of the fluid is shown in Figure 10. Buckling occurs after the viscous fluid impacts the rigid bottom plate.

DOI: 10.1002/fld



(a) Center of Mass, case $\#1$

(b) Center of Mass, case $\#2$

(c) Rise Velocity, case $\#1$

(d) Rise Velocity, case $\#2$

(e) Zero level-set at $t = 3$, case $\#1$

(f) Zero level-set at $t = 3$, case $\#2$

Figure 8. Rising bubble test cases #1 and #2. Our simulations are within the range of the benchmark. (a) Center of mass, case #1, (b) Center of mass, case #2, (c) rise velocity, case #1, (d) rise velocity, case #2, (e) zero level set at $t = 3$, case #1, and (f) zero level set at $t = 3$, case #2.

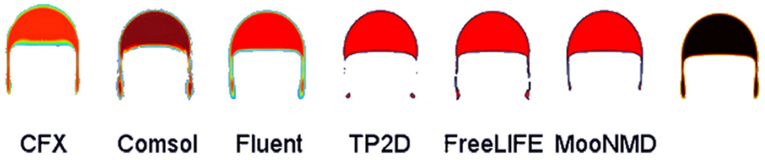


Figure 9. Rising bubble case #2 at $t = 3$. Different algorithms described in [23], and our simulation in the right-most panel. The shapes of the bubbles are qualitatively similar. The left figures are courtesy of S. Turek [34].

We also show in Figure 11 a three-dimensional simulation. The computational domain is $\Lambda = (0, 0.008\text{m})^3$, the initial diameter of the jet is 0.0004 m and the inflow velocity is $1.75\,\text{ms}^{-1}$. The viscosity and density of the fluid are those of silicone oil: $\rho^+ = 960\,\text{kg}\,\text{m}^{-3}$, $\mu^+ = 5\text{Pa s}$. The viscosity and density of the ambient fluid are those of air: $\rho^- = 1.2\,\text{kg}\,\text{m}^{-3}$, $\mu^- = 2 \times 10^{-5}\text{Pa s}$. Surface tension is accounted for, $\sigma = 0.021\ \text{Nm}^{-1}$. The no-slip boundary condition $(\mathbf{u} = \mathbf{0})$ is

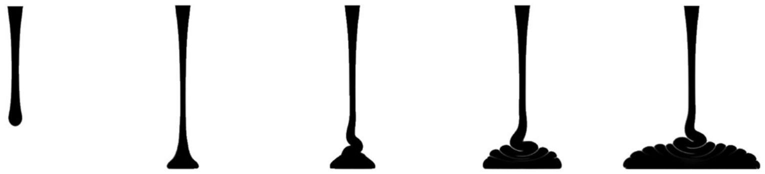


Figure 10. (From left to right) Time evolution of a jet of very viscous liquid inside a cavity filled with air. Buckling occurs when the liquid hits the floor.

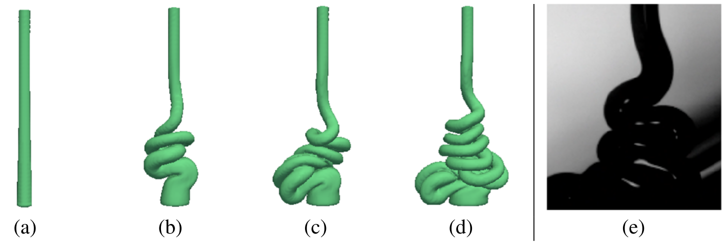


Figure 11. (a)–(e) Time evolution of a three-dimensional jet of Silicone oil falling in a cavity filled with air at time (a) $t = 0.00501$, (b) $t = 0.0209$, (c) $t = 0.031$, and (d) $t = 0.048998$ s. For comparison, (e) shows the shape of the Silicone oil jet obtained in laboratory with the same physical conditions. Our numerical simulations are qualitatively in agreement with the physical experiment.

imposed at the bottom of the box. The inflow boundary is the disk $\sqrt{(x-0.004)^2+(y-0.004)^2} < 0.0002$ at the top boundary ($z = 0.008$). Open boundary conditions $(2\mu\nabla^S\mathbf{u} - pI)\boldsymbol{\nu} = 0$ are applied on the rest of the boundary. The results are obtained with adaptive mesh refinement with the minimum cell diameter 6.25×10^{-5}. The time steps follow the CFL restriction (27) and the numerical constants are given in Table I.

5. NUMERICAL SIMULATIONS OF BOUNCING JETS

We now use our algorithm to predict the bouncing effect of jets of Newtonian (Section 5.1) and non-Newtonian (Section 5.2) fluids. In both cases, the formation of a thin layer of air between the jet and the bulk of the fluid is a critical ingredient to observe the bouncing effect. The adaptive mesh refinement strategy adopted in our algorithm allows to capture this thin layer with a reasonable number of degrees of freedom.

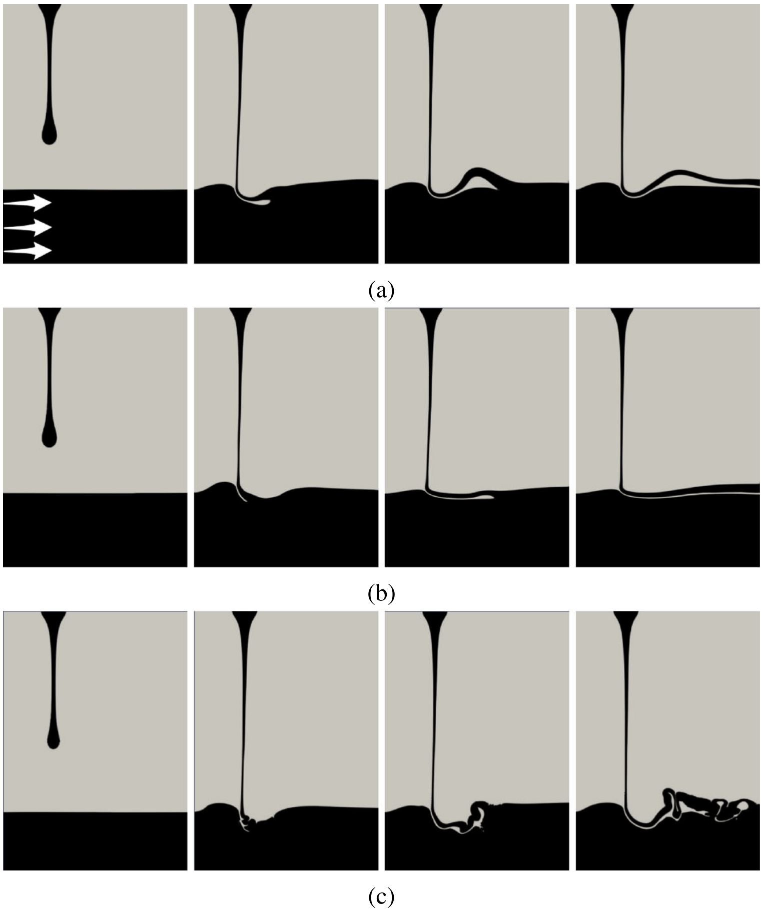
5.1. Two-dimensional Newtonian bouncing jets

We start with a Newtonian fluid falling into a translating bath as in the experiment proposed in [1]. The fluid is a silicone oil with viscosity $\mu^+ = 0.25$ Pa s and density $\rho^+ = 960$ kg m^{-3}. The ambient fluid is air: $\rho^- = 1.2$ kg m^{-3} and $\mu^- = 2 \times 10^{-5}$ Pa s. The computational domain is $\Lambda := (0, 0.04 \text{ cm}) \times (0, 0.06 \text{ cm})$. The radius of the incoming jet is 0.25 cm; its velocity is $\mathbf{u} = (0, -5 \text{ cm s}^{-1})$ on $\{(x,y) \in \partial\Lambda : y = 4, |x - 0.01| < 0.0025\}$. The region $\{(x,y) \in \Lambda : 0 < y < 0.02\}$ is filled by the same fluid, called the 'bath', and it moves to the right with a horizontal velocity $V_{\boldsymbol{B}}$ to be specified later. Slip-boundary condition (2c) is imposed at the bottom of the cavity, and the open boundary condition $(2\mu\nabla^S\mathbf{u} - pI)\mathbf{u} = 0$ is applied to the rest of the boundary

$$\begin{aligned} &\{(x,y) \in \partial\Lambda : y = 4, |x - 0.01| \geqslant 0.0025\} \\ &\cup\{(x,y) \in \partial\Lambda : x = 0, y \geqslant 0.02\} \cup \{(x,y) \in \partial\Lambda : x = 0.04\}. \end{aligned}$$

The values of the numerical constants are those given in Table I. The minimum cell diameter resulting from the adaptive mesh refinement strategy is 7.8125×10^{-5} cm. The time steps are chosen to satisfy the CFL restriction (27).

As already noted in [1], there is a range of velocities for which bouncing occurs, and the jet slides along the surface of the bath when the horizontal velocity of the bath is too high. This is illustrated


(a)

(b)

(c)

Figure 12. Newtonian bouncing jet (from left to right). The white arrow indicates the translation direction of the bath. (a) Bath velocity $V_B = 8\,\mathrm{cm\,s^{-1}}$ and surface tension coefficient $\sigma = 21\,\mathrm{mN\,m^{-1}}$; the incoming jet bounces away from the bath, and we observe the apparition of an air layer between the jet and the bath. (b) Bath velocity $V_B = 25\,\mathrm{cm\,s^{-1}}$ and surface tension coefficient $\sigma = 21\,\mathrm{mN\,m^{-1}}$; in this case, the bath velocity is too large and the jet slides along the bath surface; (c) bath velocity $V_B = 8\,\mathrm{cm\,s^{-1}}$ but without surface tension $\sigma = 0\,\mathrm{mN\,m^{-1}}$; compare with (a).

in Figure 12(a) and (b): the jet bounces when $V_B = 8\,\mathrm{cm\,s^{-1}}$ (Figure 12(a)), but it slides when $V_B = 25\,\mathrm{cm\,s^{-1}}$ (Figure 12(b)). Note that it is necessary to include the surface tension to keep the jet stable after impinging on the free surface as illustrated in Figure 12(c) where a simulation without surface tension is presented. Observe finally that all our numerical simulations show that a thin layer of air is formed between the jet and the bath each time the jet bounces.

5.2. *Kaye effect*

The Kaye effect is the name given to the bouncing jet phenomenon when the bath is stationary. This effect has been observed to occur only with non-Newtonian fluids. It is now recognized that shear-thinning viscosity is a critical component of the Kaye effects, see, for example, [4–6]. Following [5], we adopt the model of [9] in the rest of the paper

$$\mu(\gamma) = \mu_\infty + \frac{\mu_0 - \mu_\infty}{1 + \left(\dfrac{\gamma}{\gamma_c}\right)^n}, \tag{52}$$

where μ_0 is the viscosity at zero-shear stress, μ_∞ is the limiting viscosity for large stresses, γ is the Frobenius norm of the rate-of-strain tensor

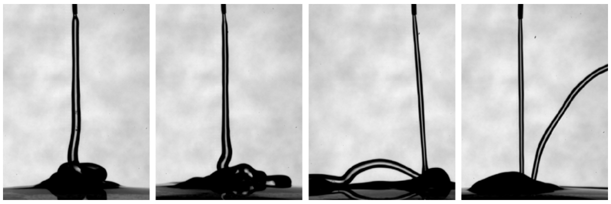
$$\gamma := \|\nabla^S \mathbf{u}\| := \left(\sum_{i=1}^{d} \sum_{j=1}^{d} \frac{1}{4} \left(\frac{\partial \mathbf{u}_j}{\partial \mathbf{x}_i} + \frac{\partial \mathbf{u}_i}{\partial \mathbf{x}_j} \right)^2 \right)^{\frac{1}{2}},$$

and γ_c, n are two additional parameters. As a benchmark, we consider a commercial shampoo for which the shear-thinning constants corresponding to the aforementioned model (52) have been identified experimentally in [7]. Recall that viscosities cannot be measured directly but are deduced from velocity and displacement measurements so as to match in some least-squares sense a behavior conjectured a priori. The parameters $\mu_0, \mu_\infty, \gamma_c, n$ corresponding to the model (52) obtained in [7] are

$$\mu_0 = 5.7\,\text{Pa s}, \quad \mu_\infty = 10^{-3}\,\text{Pa s}, \quad \gamma_c = 15\,\text{s}^{-1}, \quad \text{and} \quad n = 1. \tag{53}$$

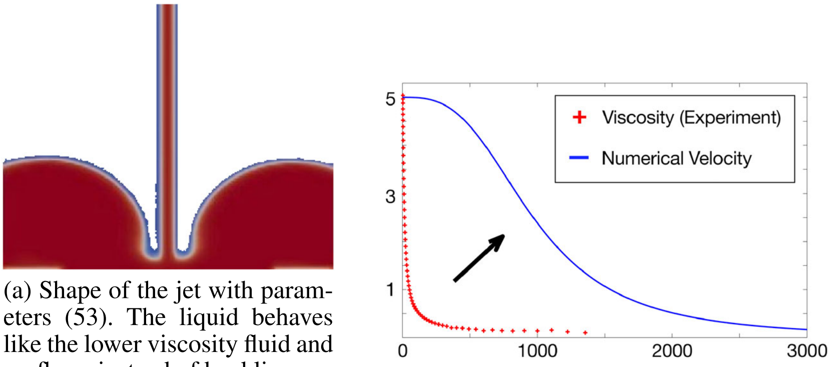
We show in Figure 13 snapshots of experiments performed with the shampoo poured at different flow rates. It was observed unambiguously in [7] that the jet slides on a lubricating air layer.

In order to reproduce qualitatively the preceding experiments, we consider the two-dimensional computational domain $\Lambda = (0.496\,\text{m}, 0.594\,\text{m}) \times (0, 0.016\,\text{m})$. The no-slip boundary condition $(\mathbf{u} = \mathbf{0})$ is imposed at the bottom of the computational domain, and an inflow boundary condition is imposed on the disk $\{(x, y) \mid |x - 0.5| < 0.00021875\}$ on the top of the box. This correspond to a jet of radius 0.4375 mm. The inflow velocity is taken to be 1.75 m s^{-1}. The open boundary condition $(2\mu\nabla^S\mathbf{u} - pI)\mathbf{u} = 0$ is applied on all the other boundaries. The numerical constants used in the simulations are listed in Table I. The minimum mesh size attainable by adaptive mesh refinement is 3.125×10^{-5} m, and time steps are chosen to comply with the CFL restriction (27). The physical parameters chosen for the fluid are $\rho^+ = 1020\,\text{kg m}^{-3}$, the shear-thinning viscosity constants are



(a) 5 ml/min (b) 6 ml/min (c) 7 ml/min (d) 8 ml/min

Figure 13. Shampoo poured at different flow rates (a) 5 (b) 6; (c) 7; and (d) 8 mL min^{-1}; No bouncing is observed at low flow rates (a) and (b). However, the Kaye effect is observed at higher flow rates (c) and (d).



(a) Shape of the jet with parameters (53). The liquid behaves like the lower viscosity fluid and flows instead of buckling.

(b) Redefined viscosity

Figure 14. (Left) Falling jet of liquid described by the parameters given in (53). The shear-thinning effect is too strong, and the fluid becomes instantly water-like when hitting the bottom of the cavity. (Right) Viscosity versus shear for the parameters (53) (dotted line) and (54) (solid line). (a) Shape of the jet with parameters (53). The liquid behaves like the lower viscosity fluid and flows instead of buckling. Shape of the jet with parameters (53). The liquid behaves like the lower viscosity fluid and flows instead of buckling, (b) redefined viscosity.

DOI: 10.1002/fld

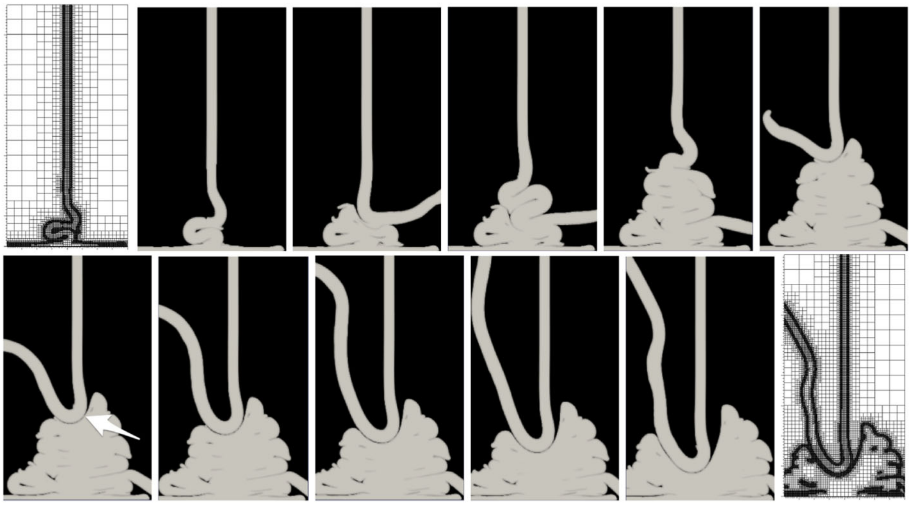


Figure 15. (Left to Right, Top to Bottom) Numerical simulation of the Kaye effect with adaptive meshes (from left to right and top to bottom). The viscosity parameters are given in (54). The first and last frames illustrate the adaptive subdivision generated by the adaptive strategy. An air layer appears in the last frame in the first row and is fully developed in the first frame of the second row. The apparition of the air layer coincides with the beginning of the bouncing phenomenon.

provided in (54). We take $\rho^- = 1.2\,\mathrm{kg\,m^{-3}}$ and $\mu^- = 2 \times 10^{-5}\,\mathrm{Pa\,s^{-5}}$ for the air. Surface tension is applied at the fluid/air interface with the surface tension coefficient $\sigma = 0.03\,\mathrm{N\,m^{-1}}$.

After many numerical experiments, it turned out that the aforementioned physical parameters did not give any Kaye effect. Our interpretation is that the shear-thinning effect given by the set of parameters (53) is too strong; with these coefficients, the fluid instantly becomes water like when hitting the bottom of the cavity (Figure 14(a)). Several explanations for this mismatch are plausible: (i) as mentioned earlier, the parameters (53) are measured under the assumption that the shear-thinning law follows the Cross model; (ii) the shear is over-predicted by our algorithm; (iii) the air layer width for this range of parameters is too thin to be captured by the algorithm; and (iv) a fundamental component is missing in our mathematical model. These observations have lead us to consider a different set of shear-thinning parameters requiring a larger shear for a notable reduction of the viscosity and a smoother transition from the maximum to the minimum values of the viscosity (Figure 14(b)). The parameters that we now consider are the following:

$$\mu_0 = 5.7\,\mathrm{Pa\,s^{-3}}, \quad \mu_\infty = 10^{-3}\,\mathrm{Pa\,s}, \quad \gamma_c = 970\,\mathrm{s^{-1}} \quad \text{and} \quad n = 3. \tag{54}$$

This set of parameter produces the Kaye effect as shown in Figure 15. Here again, we notice that there is a very thin layer of air between the bouncing jet and the rest of the fluid, thereby adding to the large body of evidence pointing at the importance of air layers in bouncing jets. Let us finally mention that these computations show also that mesh adaptivity is critical to reproduce numerically the Kaye effect.

6. CONCLUSION

Newtonian and non-Newtonian (Kaye effect) bouncing jets have been successfully reproduced numerically. A level set representation has been employed to track the interface between two immiscible incompressible fluids modeled by the incompressible Navier–Stokes equations with variable density and viscosity.

In agreement with experimental results, we have found that the creation of a thin lubricating layer of air between the jet and the rest of the fluid is a critical aspect of this phenomenon. Capturing numerically such a thin layer was possible upon designing algorithms capable of efficiently reproducing the evolution of a level set function with a sharp transition layer. Among the numerical tools that helped us achieve this goal are (i) a reinitialization algorithm based on a hyperbolic tangent filter; (ii) an (on the fly) entropy viscosity stabilization technique coupled with an explicit SSP Runge–Kutta 3 discretizations in time; and (iii) an adaptive finite element method tailored to interface problems described by level set functions. Several two-dimensional and three-dimensional numerical benchmark problems have been provided to validate our method and thereby gave credibility to our findings.

ACKNOWLEDGEMENTS

The experiments reported in the paper were carried out at the High Speed Fluid Imaging Laboratory of S. Thoroddsen at KAUST during two visits of S. L. The authors would like to express their gratitude to S. Thoroddsen and E. Li for their help in conducting these experiments. W. Bangerth availability and his constant help in the implementation of our algorithms with the deal.II library is acknowledged. The authors acknowledge the Texas A&M University Brazos HPC cluster that contributed to the research reported here.

This work has been supported by the National Science Foundation through grant DMS-1254618 and the King Abdullah University of Science and Technology award KUS-C1-016-04.

REFERENCES

1. Thrasher M, Jung S, Pang YK, Chuu CP, Swinney HL. Bouncing jet: a Newtonian liquid rebounding off a free surface. *Physical Review E* 2007; **76**:056319.
2. Lockhart T. Bouncing of Newtonian liquid jets, 2014. (Available from: https://www.uwec.edu/Physics/research/lockhartjets.htm) [Accessed on 30 January 2013].
3. Kaye A. A bouncing liquid stream. *Nature* 1963; **197**:1001.
4. Collyer A, Fischer PJ. The Kaye effect revisited. *Nature* 1976; **261**:682.
5. Versluis M, Blom C, van der Meer D, van der Weele K, Lohse D. Leaping shampoo and the stable Kaye effect. *Journal of Statistical Mechanics: Theory and Experiment* 2006; **2006**(07):P07007.
6. Binder JM, Landig AJ. The Kaye effect. *European Journal of Physics* 2009; **30**(6):S115.
7. Lee S, Li EQ, Marston JO, Bonito A, Thoroddsen ST. Leaping shampoo glides on a lubricating air layer. *Physical Review E* 2013; **87**:061001.
8. Ochoa J, Guerra C, Stern C. New experiments on the Kaye effect. In *Experimental and Theoretical Advances in Fluid Dynamics, Environmental Science and Engineering*, Klapp J, Cros A, Velasco Fuentes O, Stern C, Rodriguez Meza MA (eds). Springer: Berlin Heidelberg, 2012; 419–427.
9. Cross MM. Rheology of non-Newtonian fluids: a new flow equation for pseudoplastic systems. *Journal of Colloid Science* 1965; **20**(5):417–437.
10. Chorin AJ. Numerical solution of the Navier–Stokes equations. *Mathematics of Computation* 1968; **22**:745–762.
11. Témam R. Sur l'approximation de la solution des équations de Navier–Stokes par la méthode des pas fractionnaires. II. *Archive for Rational Mechanics and Analysis* 1969; **33**:377–385.
12. Osher S, Sethian JA. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics* 1988; **79**(1):12–49.
13. Bonito A, Guermond JL, Popov B. Stability analysis of explicit entropy viscosity methods for non-linear scalar conservation equations. *Mathematics of Computation* 2014; **83**(287):1039–1062.
14. Ville L, Silva L, Coupez T. Convected level set method for the numerical simulation of fluid buckling. *International Journal for Numerical Methods in Fluids* 2011; **66**(3):324–344.
15. Guermond JL, Pasquetti R, Popov B. Entropy viscosity method for nonlinear conservation laws. *Journal of Computational Physics* 2011; **230**(11):4248–4267.
16. Gottlieb S, Shu CW, Tadmor E. Strong stability-preserving high-order time discretization methods. *SIAM Review* 2001; **43**(1):89–112.
17. Shu CW, Osher S. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics* 1988; **77**(2):439–471.
18. Guermond JL, Shen J. On the error estimates for the rotational pressure-correction projection methods. *Mathematics of Computation* 2004; **73**(248):1719–1737 (electronic).
19. Guermond JL, Minev P, Shen J. Error analysis of pressure-correction schemes for the time-dependent Stokes equations with open boundary conditions. *SIAM Journal on Numerical Analysis* 2005; **43**(1):239–258 (electronic).
20. Guermond JL, Salgado A. A splitting method for incompressible flows with variable density based on a pressure Poisson equation. *Journal of Computational Physics* 2009; **228**(8):2834–2846.
21. Bonito A, Guermond JL, Lee S. Modified pressure-correction projection methods: Open boundary and variable time stepping. In *Numerical Mathematics and Advanced Applications - ENUMATH 2013*, vol. 103, Lecture Notes in Computational Science and Engineering, 2015; 623–631.
22. Bänsch E. Finite element discretization of the Navier–Stokes equations with a free capillary surface. *Numerische Mathematik* 2001; **88**(2):203–235.

DOI: 10.1002/fld

23. Hysing S, Turek S, Kuzmin D, Parolini N, Burman E, Ganesan S, Tobiska L. Quantitative benchmark computations of two-dimensional bubble dynamics. *International Journal for Numerical Methods in Fluids* 2009; **60**(11): 1259–1288.
24. Dziuk G, Elliott CM. An Eulerian approach to transport and diffusion on evolving implicit surfaces. *Computer and Visualization in Science* 2010; **13**(1):17–28.
25. Gilbarg D, Trudinger NS. *Elliptic Partial Differential Equations of Second Order*, Classics in Mathematics. Springer-Verlag: Berlin, 2001. Reprint of the 1998 edition.
26. Engquist B, Tornberg AK, Tsai R. Discretization of Dirac delta functions in level set methods. *Journal of Computational Physics* 2005; **207**(1):28–51.
27. Tornberg AK. Interface tracking methods with application to multiphase flows, Royal Institute of Technology, Doctoral Dissertation, 2000.
28. Bonito A, Nochetto RH. Quasi-optimal convergence rate of an adaptive discontinuous Galerkin method. *SIAM Journal on Numerical Analysis* 2010; **48**(2):734–771.
29. Bangerth W, Hartmann R, Kanschat G. deal.II — a general purpose object oriented finite element library. *ACM Transactions on Mathematical Software* 2007; **33**(4):24/1–24/27.
30. Bangerth W, Heister T, Heltai L, Kanschat G, Kronbichler M, Maier M, Turcksin B, Young TD. The `deal.ii` library, version 8.1. *arXiv preprint*, 2013.
31. Gabriel E, Fagg G. E, Bosilca G, Angskun T, Dongarra JJ, Squyres JM, Sahay V, Kambadur P, Barrett B, Lumsdaine A, Castain RH, Daniel DJ, Graham RL, Woodall TS. Open MPI: goals, concept, and design of a next generation MPI implementation. *Proceedings, 11th European PVM/MPI Users' Group Meeting*, Budapest, Hungary, 2004; 97–104.
32. Burstedde C, Wilcox LC, Ghattas O. `p4est`: scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM Journal on Scientific Computing* 2011; **33**(3):1103–1133.
33. Zalesak ST. Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of Computational Physics* 1979; **31**(3):335–362.
34. Rising bubble benchmark problems, 2014. (Available from: http://www.featflow.de/en/benchmarks/cfdbenchmarking/bubble.html) [Accessed on January 2015].
35. Bonito A, Picasso M, Laso M. Numerical simulation of 3D viscoelastic flows with free surfaces. *Journal of Computational Physics* 2006; **215**(2):691–716.
36. Tome M, McKee S. Numerical simulation of viscous flow: buckling of planar jets. *International Journal for Numerical Methods in Fluids* 1999; **29**(6):705–718.

DOI: 10.1002/fld