A Spherical Kernel for the Finite Volume Particle Method and Application to Surface Tension

Audrey Maertens, Ebrahim Jahanbakhsh, François Avellan
Laboratory for Hydraulic Machines
École Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland
audrey.maertens@epfl.ch, ebrahim.jahanbakhsh@epfl.ch, francois.avellan@epfl.ch

***Abstract*—We have recently developed a conservative finite volume particle method (FVPM) that can efficiently model 2D and 3D fluid flow with free-surfaces and complex geometry, as well as fluid-structure interaction. In this paper we present a new FVPM formulation that features spherical kernel support in place of the original cubic support. Spherical kernels have no directionality and result in smooth interactions between particles, which improves the accuracy and robustness of the method. Building on the spherical kernel FVPM, we introduce a new surface tension model. The formulation, derived from a physical model, is based on macroscopic symmetrical particle-particle interaction forces and results in a stable surface tension force with no ad-hoc parameters.**

I. Introduction

The Finite Volume Particle Method (FVPM) is a meshless arbitrary Lagrangian-Eulerian (ALE) method introduced by Hietel et al. [1] in 2000 for compressible flows. This method has since been used in a wide range of applications such as incompressible flows [2], solid mechanics [3], fluid-structure interactions [4], [5], free-surface flows in Pelton turbines [6], [7], and silt erosion [8].

The FVPM features many of the attractive properties of both particle methods, such as Smoothed Particle Hydrodynamics (SPH) [9], and conventional mesh-based Finite Volume Methods (FVM) [10]. In FVPM, like in SPH, computational nodes usually move with the material velocity, which is compatible with the Lagrangian form of the equation of motion. This enables the method to handle moving interface problems like free-surface flows, without dealing with mesh deformation or tangling. Moreover, FVPM does not require mesh generation which is a costly stage in simulation of flows with complex geometries.

Similarly to FVM, FVPM is locally conservative and consistent, regardless of any variation in volume sizes. In fact, FVPM can be interpreted as a generalization of conventional mesh-based FVM [11]. In FVM, the computational domain is partitioned into finite control volumes with defined surfaces. The area vector of the surfaces is used as a weight for the flux exchanged between the control volumes. In FVPM, control volumes are replaced by overlapping particles and the exchange occurs through the interfaces defined by overlapping regions. For each pair of overlapping particles, two interaction vectors are defined and their difference is analogous to the area vector in FVM. The interaction vectors can be computed by either numerical or exact integration. Numerical integration is costly and approximate, and is mostly used for bell-shaped kernels [2], [12], [13]. Exact integration, introduced by Quinlan et al. [5], is based on top-hat kernels, resulting in a reasonable compromise between efficiency and accuracy. This method was originally developed for 2D computations using circular and rectangular top-hat kernels [14]. Recently, Jahanbakhsh et al. [15] developed a 3D FVPM formulation which features exact integration based on cubic-supported top hat kernels.

However, employing rectangular top-hat kernels such as square or cubic-supported kernels in FVPM results in two important issues. First of all, these kernels have directionality, which destroys volume conservation in pure rigid body rotation and accordingly impairs the accuracy of the method. Secondly, the particle interaction can result in hard contact which causes high-frequency errors. Both issues disappear with circular or spherical-supported top-hat kernels.We present in this paper a technique which uses spherical top-hat kernels for exact integration of the FVPM interaction vectors. This technique is based on an innovative surface partitioning algorithm relying on logical set operations and precise area evaluation to robustly handle complicated particle intersections.

In addition to the intrinsic advantages of spherical kernel support, the new formulation can be used as the base of an innovative surface tension model. The model uses the interaction vectors calculations to compute a symmetrical particle-particle surface tension force. Unlike most symmetrical surface tension models in SPH [16], the formulation is derived from a macroscopic physical model. The model uses only surface tension and equilibrium contact angle as input parameters and the resulting force is stable and consistant while conserving momentum. In order to illustrate the performance of the new surface tension model, we present simulations of a drop in air and falling onto plane of various wetting conditions.

This paper is organized as follows. Section II summarizes the FVPM equations for fluid flow computations. In section III, we present our approach to efficiently compute the interaction vectors with spherical support kernel, and section IV presents the new surface tension model. Finally, applications of the new FVPM formulation and surface tension model to water drops

are presented in section V.

II. Governing Equations and FVPM Formulation

A. Governing Equations

The equations of motion for isothermal and weakly compressible flows are derived from the mass and linear momentum conservation laws

$$\frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot \boldsymbol{F}\left(\boldsymbol{U}\right) = 0 \tag{1}$$

where $\boldsymbol{U}$ represents the conserved variables and $\boldsymbol{F}$ represents the flux functions. For fluid flow equations, the corresponding variables and flux functions are respectively $\boldsymbol{U} = \begin{pmatrix} \rho \\ \rho \boldsymbol{C} \end{pmatrix}$ and $\boldsymbol{F} = \begin{pmatrix} \rho \boldsymbol{C} \\ \rho \boldsymbol{C} \otimes \boldsymbol{C} - \boldsymbol{s} + p\mathbf{I} \end{pmatrix}$, where $\boldsymbol{C}$ represents the fluid velocity, ρ the density, p the pressure, and $\boldsymbol{s}$ the deviatoric stress tensor. To close the system of equations, the Murnaghan-Tait equation of state is used for water [17]

$$p = \frac{\rho_\circ a^2}{\gamma}\left(\left(\frac{\rho}{\rho_\circ}\right)^\gamma - 1\right) \tag{2}$$

where a is the numerical speed of sound, $\rho_\circ$ is the reference density and γ is a constant coefficient that we set to $\gamma = 7$. In weakly compressible flow simulations, the numerical speed of sound a is ten times the maximum fluid velocity, which significantly reduces the computational cost [18].

B. The FVPM Formulation

The FVPM formulation for conservation laws (1) reads

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{U}_i V_i\right) = &\sum_j \left(\boldsymbol{U}_{ij} \otimes \dot{\boldsymbol{x}}_{ij} - \boldsymbol{F}_{ij}\right) \cdot \boldsymbol{\Delta}_{ij} \\ &+ \left(\boldsymbol{U}_b \otimes \dot{\boldsymbol{x}}_b - \boldsymbol{F}_b\right) \cdot \boldsymbol{B}_i\end{aligned} \tag{3}$$

and

$$\frac{\mathrm{d}V_i}{\mathrm{d}t} = \sum_j \dot{\boldsymbol{x}}_{ij} \cdot \boldsymbol{\Delta}_{ij} + \dot{\boldsymbol{x}}_b \cdot \boldsymbol{B}_i \tag{4}$$

with

$$\boldsymbol{\Delta}_{ij} = \boldsymbol{\Gamma}_{ij} - \boldsymbol{\Gamma}_{ji} \tag{5}$$

$$\dot{\boldsymbol{x}}_{ij} = \left(\dot{\boldsymbol{x}}_j \cdot \boldsymbol{\Gamma}_{ij} - \dot{\boldsymbol{x}}_i \cdot \boldsymbol{\Gamma}_{ji}\right) \frac{\boldsymbol{\Delta}_{ij}}{\boldsymbol{\Delta}_{ij} \cdot \boldsymbol{\Delta}_{ij}} \tag{6}$$

$$\boldsymbol{B}_i = -\sum_j \boldsymbol{\Delta}_{ij} \tag{7}$$

where $\boldsymbol{U}_i$ is the conserved variable of ith particle, V_i is its volume, $\boldsymbol{U}_{ij}$ and $\boldsymbol{F}_{ij}$ are the conserved variable and flux function at the interface of particles i and j, respectively, whereas $\dot{\boldsymbol{x}}_{ij}$ is the velocity at which the interface moves. Similarly, $\boldsymbol{U}_b$, $\boldsymbol{F}_b$ and $\dot{\boldsymbol{x}}_b$ are the conserved variable, flux function and particle velocity at the boundary. $\boldsymbol{\Delta}_{ij}$ and $\boldsymbol{B}_i$ are the vectors which weight exchanged fluxes between particles and with the boundary, respectively. These vectors are computed from interaction vectors $\boldsymbol{\Gamma}_{ij}$ which read

$$\boldsymbol{\Gamma}_{ij} = \int_\Omega \frac{\psi_i \nabla W_j}{\sigma}\,\mathrm{d}V. \tag{8}$$

In (8), ψ_i denotes the Shepard shape function for the i^{th} particle defined as

$$\psi_i\left(\boldsymbol{x}\right) = \frac{W_i\left(\boldsymbol{x}\right)}{\sigma\left(\boldsymbol{x}\right)} \tag{9}$$

where W_i is a kernel function,

$$W_i\left(\boldsymbol{x}\right) = W\left(\boldsymbol{x} - \boldsymbol{x}_i, h_i\right) \tag{10}$$

and σ is the kernel summation

$$\sigma\left(\boldsymbol{x}\right) = \sum_j W_j\left(\boldsymbol{x}\right)\ . \tag{11}$$

$W_i\left(\boldsymbol{x}\right)$ is defined as zero outside Ω_i, the support of particle i. Particle volume V_i is defined as

$$V_i = \int_\Omega \psi_i\,\mathrm{d}V\ . \tag{12}$$

In (10), h_i is known as the smoothing length of particle i, and defines the particle size and hence the spatial resolution of the scheme. It is worth mentioning that, as long as the sampling point $\boldsymbol{x}$ is selected within the particles smoothing support, $\sigma\left(\boldsymbol{x}\right) > 0$, and accordingly the shape function (9) is defined at that position. Since the computational domain is defined by the union of the particles smoothing support, the shape function does not need to be defined outside of this union. The readers are referred to [15] for the formulas derivation details.

III. Computation of Interaction Vectors for Spherical Particles

To solve (3), the interaction vector $\boldsymbol{\Gamma}_{ij}$ must be computed according to (8). For conventional bell-shaped kernels, this integral is difficult or impossible to exactly evaluate. The alternative approach is to use quadrature rules which are approximate and costly [1], [2]. Until recently, the computational needs of this approach precluded the use of FVPM for 3D applications.

Quinlan and Nestor [5] and Quinlan et al. [14] introduced top-hat kernels with circular and rectangular supports to compute the integrals exactly and efficiently in 2D. The top-hat kernel is defined as

$$W_i\left(\boldsymbol{x}\right) = \begin{cases} 1 & \boldsymbol{x} \in \Omega_i \\ 0 & \text{otherwise} \end{cases}. \tag{13}$$

This kernel choice enables integrals over the particle support volume to be reduced to integrals over the bounding surface of the particle support. According to Jahanbakhsh et al. [15], computation of the interaction vector $\boldsymbol{\Gamma}_{ji}$ requires that the supporting border of ith particle, $\partial\Omega_i$, be partitioned into subsurfaces of constant σ values, which are termed elementary surfaces. Then, $\boldsymbol{\Gamma}_{ji}$ is computed exactly as

$$\boldsymbol{\Gamma}_{ji} = -\sum_{\mathcal{P} \in (\Omega_j \cap \partial\Omega_i)} \left(\frac{\boldsymbol{S}_\mathcal{P}}{\sigma_\mathcal{P}^+ \sigma_\mathcal{P}^-}\right) \tag{14}$$

where $\boldsymbol{S}_\mathcal{P}$ denotes the area vector of the elementary surface $\mathcal{P}$, Ω_j denotes the supporting volume of jth particle, while

$\sigma_{\mathcal{P}}^{+}$ and $\sigma_{\mathcal{P}}^{-}$ denote the kernel summations outside and inside Ω_i, respectively.

Equation (14) is valid for top-hat kernels regardless of the particle support shape. Since $W_i(\boldsymbol{x}) = 1$ inside particle i, $\sigma(\boldsymbol{x})$ for a top-hat kernel is simply the number of particles covering the point $\boldsymbol{x}$.

It follows that

$$\sigma_{\mathcal{P}} = \sigma_{\mathcal{P}}^{-} = \sigma_{\mathcal{P}}^{+} + 1 \tag{15}$$

and, in accordance with Quinlan et al. [14], (14) simplifies to

$$\boldsymbol{\Gamma}_{ji} = \sum_{p \in (\Omega_j \cap \partial\Omega_i)} \boldsymbol{S}_{\mathcal{P}} \left(\frac{1}{\sigma_{\mathcal{P}} + 1} - \frac{1}{\sigma_{\mathcal{P}}} \right). \tag{16}$$

The key innovations of the present work is the exact computation of particle interaction vectors $\boldsymbol{\Gamma_{ij}}$ for spherical particles using top-hat kernels. The first stage is a surface partitioning algorithm which identifies the elementary surfaces formed by sphere intersections and computes kernel summations σ for each elementary surface (i.e. the number of particles inside which the surface lies).

A. Surface Partitioning

In this algorithm, we compute all the elementary surfaces generated on $\partial\Omega_i$ by the neighbors of particle i. The four steps of the surface partitioning algorithm are summarized bellow (see [19] for more details).

1) Construction of spherical caps: A spherical cap is the region of a sphere which lies on either side of a given plane [20]. The intersection of a sphere $\partial\Omega_i$ with a ball Ω_j generates a spherical cap. The first step in the surface partitioning algorithm is finding all the spherical caps generated after intersecting the surface of ith particle with its neighboring particles.

2) Intersection of surface circles: Each spherical cap has a circular base which may intersect circles formed by other caps. In the second step, we check each pair of circles appearing on the surface of ith particle for intersection. This is equivalent to finding the intersection of three spherical surfaces $\partial\Omega_i$, $\partial\Omega_j$ and $\partial\Omega_k$.

3) Construction of arc sets: In the third step, we construct the indivisible arcs formed by the circles' intersections and located on $\partial\Omega_i$. Each arc is defined by two end-points and the circle of which it is a subset. In addition, we assign an integer index to each arc ranging from 1 to N_{arc}, where N_{arc} denotes the total number of arcs on $\partial\Omega_i$. The sign of the index indicates the arc direction with respect to the normal of the associated circle. Thus, the direction of the arc can be reversed by negating its index number. Every arc is a boundary of exactly two elementary surfaces, and the use of positive and negative indices will later enable the two boundaries to be represented distinctly. For the spherical surface $\partial\Omega_i$, we can construct a complex arc set $\mathcal{U} = \{-1, +1, -2, +2, -3, +3, ..., \pm N_{\text{arc}}\}$ which contains all the arcs of $\partial\Omega_i$. Similarly, we construct a complex arc set, namely $\mathcal{A}_{ij}$, for each spherical cap $\mathcal{C}_{ij}$. This set includes the positive indices of the arcs formed by the base circle of $\mathcal{C}_{ij}$, as well as positive and negative indices of the arcs located inside Ω_j.

4) Partitioning: In the fourth step, we partition the spherical surface $\partial\Omega_i$ into the elementary surfaces which are indivisible and do not overlap with each other. The main idea of the partitioning procedure is to intersect the spherical surface with all the spherical caps, one after another. After the first intersection, the sphere surface is decomposed into two subsurfaces inside and outside of the first spherical cap. For the second intersection, we find the intersections of the two subsurfaces from previous step with the second cap. This way, new subsurfaces are generated, which are either inside or outside the second cap and will be used for the next step. If we continue this procedure for all the caps, the final subsurfaces are the required elementary surfaces. To implement this idea, we apply set operations to the complex arc sets constructed in the previous step. These operations are carried out for all neighbors j where $j = 1 \dots N_{\text{cap}}$ and N_{cap} denotes the number of spherical caps or, equivalently, the number of neighboring particles. To start, we define $\mathbb{S}_0$

$$\mathbb{S}_0 = \{\mathcal{U}\} \tag{17}$$

where $\mathbb{S}_0$ is a set of arc sets that contains only $\mathcal{U}$. Then, for each $\mathcal{A}_{ij}$ representing the cap $\mathcal{C}_{ij}$, we write

$$\mathbb{S}_j = \{\mathcal{X} \cap \mathcal{A}_{ij} : \mathcal{X} \in \mathbb{S}_{j-1}\} \cup \{\mathcal{X} - \mathcal{A}_{ij} : \mathcal{X} \in \mathbb{S}_{j-1}\}. \tag{18}$$

After performing this operation N_{cap} times, every arc set in $\mathbb{S}_{N_{\text{cap}}}$ lies inside a unique set of spherical caps, and outside all others. Therefore, it is an elementary surface which cannot be further subdivided and is defined by the intersection of a unique combination of spherical caps of neighboring particles.

During the partitioning process, we can easily find the value of σ, which is constant over each elementary surface. For this purpose, we store the σ value for each subsurface $\mathcal{X}$ and initiate it to 1 for $\mathcal{X} \in \mathbb{S}_0$ at the beginning of process. Then, for each iteration of j over (18), σ values are increased by 1 for $\mathcal{X} \in \{\mathcal{X} \cap \mathcal{A}_{ij} : \mathcal{X} \in \mathbb{S}_{j-1}\}$ and remain unchanged for $\mathcal{X} \in \{\mathcal{X} - \mathcal{A}_{ij} : \mathcal{X} \in \mathbb{S}_{j-1}\}$.

B. Area Vector Computation

In order to compute the interaction vectors according to (16), the projected area vectors $\boldsymbol{S}_{\mathcal{P}}$ of the elementary surfaces constructed in the previous section are needed. However, existing formulas for spherical polygons are only valid only for spherical polygons made of great circle arcs, i.e. arcs of circles that contain a diameter of the sphere [22]. Since the elementary surfaces are bounded by small circle arcs, i.e. arcs of circles that do not contain a diameter of the sphere [21], these formulas cannot be directly applied.

To compute the area vector of an elementary surface, we define $\boldsymbol{a}$ the north pole of the sphere and construct a triangle $\triangle^s \boldsymbol{abc}$ for each arc $\widehat{\boldsymbol{bc}}$ defining the surface and compute the area vector of each triangle, as illustrated in figures 1. Here, the superscript s in $\triangle^s$ stands for small circle and denotes a triangle with one small circle arc $\widehat{\boldsymbol{bc}}$ and two great circle

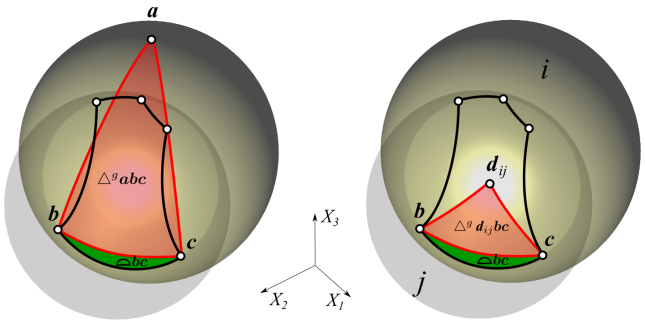

Fig. 1. On the left, the spherical triangle $\triangle^g \boldsymbol{abc}$ and the crescent $\frown \boldsymbol{bc}$ are shown in red and green, respectively. $\triangle^g \boldsymbol{abc}$ is made of the three great circle arcs passing through $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$. $\frown \boldsymbol{bc}$ is bounded by a small and a great circle arc connecting $\boldsymbol{b}$ and $\boldsymbol{c}$. On the right, the spherical triangle $\triangle^g \boldsymbol{d}_{ij}\boldsymbol{bc}$ and the crescent $\frown \boldsymbol{bc}$ are shown in red and green, respectively. $\triangle^s \boldsymbol{d}_{ij}\boldsymbol{bc}$ is made of the three great circle arcs passing through $\boldsymbol{d}_{ij}$, $\boldsymbol{b}$ and $\boldsymbol{c}$. $\frown \boldsymbol{bc}$ is bounded by a small and a great circle arc connecting $\boldsymbol{b}$ and $\boldsymbol{c}$.

arcs $\widehat{\boldsymbol{ab}}$ and $\widehat{\boldsymbol{ca}}$. If we can compute the area vectors of the triangles, summing them up will give us the area vector of the elementary surface. This technique has been used to compute the surface area of a spherical polygon in [23] and can be generalized to area vector computation.

We then decompose $\triangle^s \boldsymbol{abc}$ into a spherical triangle $\triangle^g \boldsymbol{abc}$ and a crescent $\frown \boldsymbol{bc}$, such that $\triangle^g \boldsymbol{abc}$ is made of three great circle arcs and $\frown \boldsymbol{bc}$ is surrounded by small and great circle arcs connecting $\boldsymbol{b}$ and $\boldsymbol{c}$. The superscript g in $\triangle^g$ stands for great circle and denotes a spherical triangle.

The area vector of the spherical triangle, $\boldsymbol{S}_{\triangle^g \boldsymbol{abc}}$, can be directly computed according to [23]. To compute the area vector of the crescent $\frown \boldsymbol{bc}$, we define $\boldsymbol{d}_{ij}$, the intersection of $(\boldsymbol{x}_i \boldsymbol{x}_j)$ with $\partial\Omega_i$. The area vectors $\boldsymbol{S}_{\triangle^g \boldsymbol{d}_{ij}\boldsymbol{bc}}$ and $\boldsymbol{S}_{\triangle^s \boldsymbol{d}_{ij}\boldsymbol{bc}}$ can now be computed directly, which yields the area vector of the crescent $\frown \boldsymbol{bc}$

$$\boldsymbol{S}_{\frown \boldsymbol{bc}} = \boldsymbol{S}_{\triangle^s \boldsymbol{d}_{ij}\boldsymbol{bc}} - \boldsymbol{S}_{\triangle^g \boldsymbol{d}_{ij}\boldsymbol{bc}} \tag{19}$$

Finally, the area vector of $\triangle^s \boldsymbol{abc}$ reads

$$\boldsymbol{S}_{\triangle^s \boldsymbol{abc}} = \boldsymbol{S}_{\triangle^g \boldsymbol{abc}} + \boldsymbol{S}_{\frown \boldsymbol{bc}} \tag{20}$$

To compute the area vector of any given elementary surface $\mathcal{P}$, we sum up the vector areas of all triangles $\triangle^s \boldsymbol{abc}$

$$\boldsymbol{S}_{\mathcal{P}} = \sum_{n=1}^{|\mathcal{P}|} \boldsymbol{S}_{\triangle^s \boldsymbol{a}\boldsymbol{b}_n\boldsymbol{c}_n} \tag{21}$$

where $\boldsymbol{S}_{\mathcal{P}}$ denotes the area vector of $\mathcal{P}$, $|\mathcal{P}|$ denotes the number of arcs forming $\mathcal{P}$ and $\boldsymbol{b}_n$ and $\boldsymbol{c}_n$ denote the ends of the nth arc in $\mathcal{P}$. It is worth mentioning that, since $\mathcal{P}$ can contain negative arc indices, we have to change the order of $\boldsymbol{b}_n$ and $\boldsymbol{c}_n$ for those arcs in order to have the correct sign for their corresponding areas.

IV. Surface Tension

In the context of SPH, the traditional approaches for modeling surface tension apply a force normal to the interface with an intensity determined by the given surface tension coefficient and the curvature of the interface [24]–[26]. A smooth color function is used to describe the different phases, and the interface is defined as a finite transitional band, where the color gradient does not vanish. Those approaches compute the normal for each particle using the gradient of the smoothed color field, while the curvature is computed by taking either the second derivative of the smoothed color field or the divergence of the normal field.

Such methods can accurately estimate the effects of surface tension but they require second order derivatives which are very sensitive to particle disorder and sometimes lead to significant errors. Moreover, since the forces are applied to the fluid particles as external forces in a non-symmetric way, those approaches do not conserve momentum. Finally, these methods do not readily generalize to fluid-solid interactions with wetting effects. A normal correction method is sometimes adopted to include the effects at the triple line [27], [28], but since the dynamic contact angle is usually not known [29], it only applies to quasi-static cases.

Taking advantage of the particle-based nature of SPH, several methods have been proposed to model surface tension with cohesive pair-wise forces mimicking microscopic inter-phase attractive potentials. Although the implementation of an inter-phase attractive potential is straightforward, one of the difficulties is that the resulting surface tension needs to be calibrated. Furthermore, with given parameters, the surface tension is resolution-dependent and does not converge to a fixed value with increasing resolution.

Nugent and Posch [30] used attractive forces, corresponding to the cohesive pressure in the van der Waals equation of state, to simulate surface tension in two-dimensional SPH simulations. Tartakovsky and Meakin [31] used a similar approach but instead of a van der Waals interaction applied a combination of repulsive and attractive forces within the range of the standard SPH kernel. In both methods the particular magnitude of surface tension depends on the intensity of the particle-particle interactions. These interactions are not readily available, and need to be fitted for each case in order to reproduce the desired contact angles and surface tension. Akinci et al. [16] proposed an interesting combination of cohesion, surface area minimization and adhesion which seems interesting for graphic applications but, like the previous methods, the forces rely on ad-hoc resolution-dependent parameters, which is not appropriate for scientific applications.

In most practical applications, the surface tension and contact angles are given as macroscopic input parameters, which often vary from case to case. Therefore, despite the appealing physical principle underlying these effective interaction methods, an approach that simply uses the surface tension as an input parameter is advantageous. In this section, we present a pair-wise surface tension model based on macroscopic interaction which naturally extends to model wetting effects and does not require parameter tuning. The model is based on the surface tension γ acting on a macroscopic free surface

element $\mathcal{P}$ of a fluid

$$\boldsymbol{F}^{ST}(\mathcal{P}) = -\gamma \int_{\partial\mathcal{P}} \boldsymbol{\tau}\, \mathrm{d}r \tag{22}$$

where $\boldsymbol{F}^{ST}$ is the surface tension force and $\boldsymbol{\tau}$ the local tangent to the surface $\mathcal{P}$, normal to $\partial\mathcal{P}$, and pointing inwards.

Let us consider an elementary surface $\mathcal{P} \in \partial\Omega_j$ and identify it to a macroscopic surface element of fluid. If particle i, with center at $\boldsymbol{x}_i$, is moved by $\delta\boldsymbol{x}_i$, the work done by $\boldsymbol{F}^{ST}$ on $\mathcal{P}$ is

$$\begin{aligned}\delta W_i(\mathcal{P}, \delta\boldsymbol{x}_i) &= -\gamma\ \delta S_{\mathcal{P}} = -\gamma\ \frac{\mathrm{d}S_{\mathcal{P}}}{\mathrm{d}\boldsymbol{x}_i} \cdot \delta\boldsymbol{x}_i \\ &= \boldsymbol{F}_i^{ST}(\mathcal{P}) \cdot \delta\boldsymbol{x}_i \end{aligned} \tag{23}$$

where $S_{\mathcal{P}}$ is the area of surface $\mathcal{P}$. Therefore, the surface tension force on particle i due to the elementary free surface $\mathcal{P}$ is

$$\boldsymbol{F}_i^{ST}(\mathcal{P}) = -\gamma\ \frac{\mathrm{d}S_{\mathcal{P}}}{\mathrm{d}\boldsymbol{x}_i} \tag{24}$$

Summing up the surface tension forces from all the elementary surfaces, the surface tension force on particle i reads

$$\boldsymbol{F}_i^{ST} = -\gamma \sum_j \sum_{\mathcal{P}\in\{\partial\Omega_j : \sigma = 1\}} \frac{\mathrm{d}S_{\mathcal{P}}}{\mathrm{d}\boldsymbol{x}_i} \tag{25}$$

Since $\mathcal{P} \in \partial\Omega_j$, the surface area gradient is given by

$$\frac{\mathrm{d}S_{\mathcal{P}}}{\mathrm{d}\boldsymbol{x}_i} = \begin{cases} \gamma \displaystyle\int_{\partial\mathcal{P}} \boldsymbol{\tau}\, \mathrm{d}r & \text{if } j \neq i, \\ -\gamma \displaystyle\int_{\partial\mathcal{P}} \boldsymbol{\tau}\, \mathrm{d}r & \text{if } j = i \end{cases} \tag{26}$$

Let us now consider $\widehat{\boldsymbol{b}_n\boldsymbol{c}_n} \in \partial\Omega_i \cap \partial\Omega_j$, an elementary arc bounding two elementary free surfaces $\mathcal{P}_i$ and $\mathcal{P}_j$. The surface tension force on particle i arising from the arc can be calculated using the arc properties calculated during the surface partitioning

$$\int_{\widehat{\boldsymbol{b}_n\boldsymbol{c}_n}} \boldsymbol{\tau}_i\, \mathrm{d}r = \hat{\boldsymbol{\tau}}_{ni} \frac{r_i}{d_{ij}} \tag{27}$$

where r_i is the radius of particle i, $d_{ij} = ||\boldsymbol{x}_j - \boldsymbol{x}_i||$, and

$$\begin{aligned}\hat{\boldsymbol{\tau}}_{ni} &= 2\sin(\alpha/2)(\boldsymbol{x}_j - \boldsymbol{m}_n) \\ &\quad + (\boldsymbol{x}_j - \boldsymbol{m}_n) \cdot \boldsymbol{n}_{ij}(\alpha - 2\sin(\alpha/2))\boldsymbol{n}_{ij}\end{aligned} \tag{28}$$

with α the central angle of the arc, $\boldsymbol{m}_n$ the middle of the arc and $\boldsymbol{n}_{ij} = (\boldsymbol{x}_j - \boldsymbol{x}_i)/d_{ij}$.

Finally, defining $\mathcal{A}_{ij}^{FS}$ the set of arcs separating free surfaces from particles i and j, the surface tension force on i from particle j reads

$$\boldsymbol{F}_{ji}^{ST} = -\gamma \sum_{n=1}^{|\mathcal{A}_{ij}^{FS}|} \left(\hat{\boldsymbol{\tau}}_{nj} \frac{r_j}{d_{ij}} - \hat{\boldsymbol{\tau}}_{ni} \frac{r_i}{d_{ij}}\right) \left(1 - \frac{1}{\eta^2}\right) \tag{29}$$

where the last parenthesis with $\eta = (r_i + r_j)/d_{\mathrm{ref}}$ is a correction factor compensating for the artificially larger surface area due to the use of spherical particles to represent a flat surface. It appears clearly from (29) that the surface tension force is a symmetric interparticle force.

Let us now consider a configuration with three interfaces: liquid-air ($\mathcal{I}_{LA}$), liquid-solid ($\mathcal{I}_{LS}$), and solid-air ($\mathcal{I}_{SA}$), with respective interfacial energy γ_{LA}, γ_{LS}, and γ_{SA}. The surface tension force on fluid particle i generalizes easily from (25)

$$\begin{aligned}\boldsymbol{F}_i^{ST} = &-\gamma_{LA} \sum_{\mathcal{P}\in\mathcal{I}_{LA}} \frac{\mathrm{d}S_{\mathcal{P}}}{\mathrm{d}\boldsymbol{x}_i} - \gamma_{LS} \sum_{\mathcal{P}\in\mathcal{I}_{LS}} \frac{\mathrm{d}S_{\mathcal{P}}}{\mathrm{d}\boldsymbol{x}_i} \\ &- \gamma_{SA} \sum_{\mathcal{P}\in\mathcal{I}_{SA}} \frac{\mathrm{d}S_{\mathcal{P}}}{\mathrm{d}\boldsymbol{x}_i}\end{aligned} \tag{30}$$

In general, the liquid-solid and solid-air interfacial energy are not known, but the equilibrium contact angle θ_C can be measured experimentally. The equilibrium contact angle is linked to the interfacial energies by

$$\gamma_{SA} - \gamma_{LS} = \gamma_{LA} \cos\theta_C \tag{31}$$

Using (31) and the property that the total area of the solid surface is constant, (30) simplifies to

$$\boldsymbol{F}_i^{ST} = -\gamma_{LA} \left(\sum_{\mathcal{P}\in\mathcal{I}_{LA}} \frac{\mathrm{d}S_{\mathcal{P}}}{\mathrm{d}\boldsymbol{x}_i} + \cos\theta_C \sum_{\mathcal{P}\in\mathcal{I}_{SA}} \frac{\mathrm{d}S_{\mathcal{P}}}{\mathrm{d}\boldsymbol{x}_i} \right) \tag{32}$$

where the derivatives are computed according to (26) and (27). In FVPM, this force is added to the right hand side of the momentum component of the conservation equations (3).

V. Applications

A. Levitating cubic drop

The first expected effect of surface tension is minimizing the surface energy, and hence produce spherical drops in the absence of gravity. However, when cohesion forces are used to model surface tension, spherical drops do not always turn into spheres [16]. Akinci et al. [16] argue that cohesion forces alone cannot reproduce the macroscopic surface tension behavior and an extra force is necessary to enforce surface area minimization.

With our model, a single force, based on macroscopic surface tension, enables proper prediction of drop shape, as illustrated in figure 2. Indeed, unlike the cohesion models, our surface tension force is derived from macroscopic considerations. The resulting force acts as a cohesion force since, if $r_i = r_j$, $\boldsymbol{F}_{ji}^{ST}$ is aligned with $\boldsymbol{n}_{ij}$. However, the summation over the neighbors indirectly computes the curvature, and the resulting surface tension force also minimizes the surface area and provides the expected pressure difference between the drop and the atmosphere.

B. Wetting effects

Next, a spherical drop impacts a horizontal plane with two different wetting conditions.

We first consider a hydrophobic plane with equilibrium contact angle $\theta_C = 140°$ (Figure 3). The surface tension force on particle i can be expressed as

$$\boldsymbol{F}_i^{ST} = -0.072 \frac{\mathrm{d}S^{LA}}{\mathrm{d}\boldsymbol{x}_i} + 0.055 \frac{\mathrm{d}S^{SA}}{\mathrm{d}\boldsymbol{x}_i}. \tag{33}$$

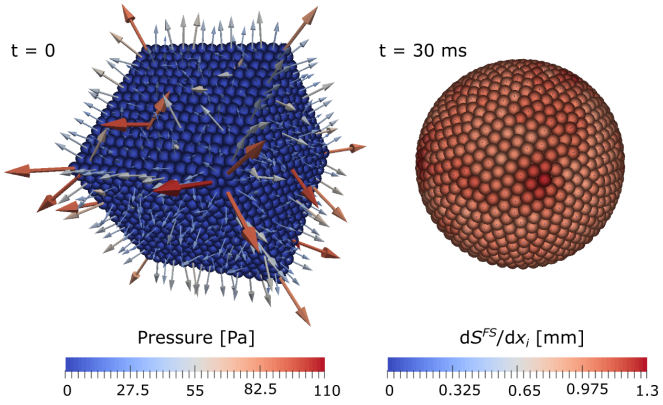

Fig. 2. Cubic drop (left) evolving into a spherical drop (right) under the action of surface tension $\gamma = 0.072$ N/m. The particles of radius $r_i = 0.183$ mm are initially randomly distributed in a cube of width 2.75 mm. The particles are colored according to their pressure and an arrow, sized and colored according to the gradient of the free surface area, is also represented for each particle. The radius of the final drop is 1.73 mm, which according to the Young-Laplace equation, would result in a pressure jump of 83 Pa.

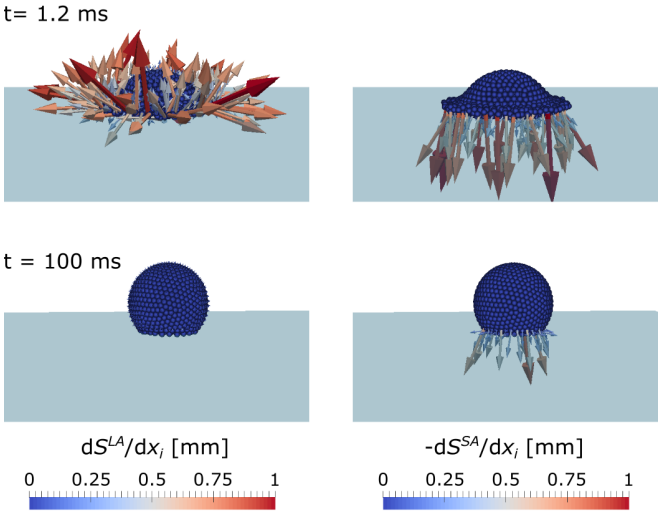

Fig. 3. Spherical water drop of radius 1.2 mm falling onto a hydrophobic plane, 1.2 ms (left) and 100 ms (right) after impact. Arrows, sized and colored according to the gradient of the liquid-air interface area (left) and solid-air interface area (right) are represented for each particle.

As the drop hits the plane, the force due to the liquid-air interface points inwards, in order to minimize the liquid-air surface area. The force due to the solid-air interface also points inward, in order to minimize the liquid-solid interface, or maximize the solid-air interface. At equilibrium, the resulting drop is mostly spherical, with a small surface contact between the drop and the plane; a configuration in which the horizontal component of both forces cancel.

When a hydrophilic plane is used, with contact angle $\theta_C = 40°$ (Figure 4), the surface tension force on particle i can be expressed as

$$\boldsymbol{F}_i^{ST} = -0.072\frac{\mathrm{d}S^{LA}}{\mathrm{d}\boldsymbol{x}_i} - 0.055\frac{\mathrm{d}S^{SA}}{\mathrm{d}\boldsymbol{x}_i}. \qquad (34)$$

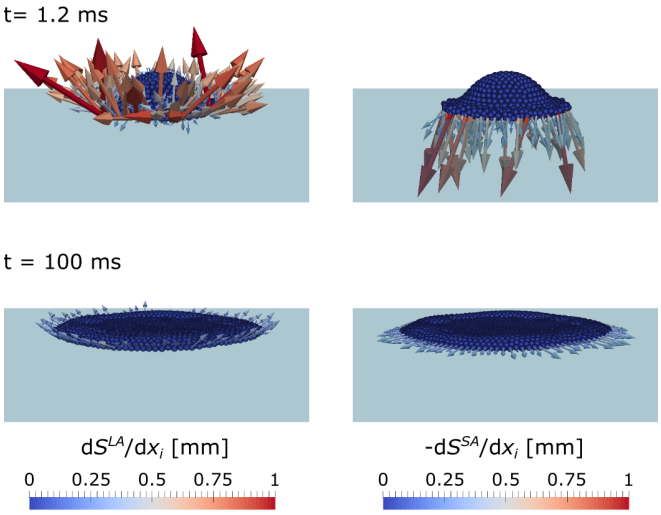

Fig. 4. Spherical water drop of radius 1.2 mm falling onto a hydrophilic plane, 1.2 ms (left) and 100 ms (right) after impact. Arrows, sized and colored according to the gradient of the liquid-air interface area (left) and solid-air interface area (right) are represented for each particle.

As the drop hits the plane, the force due to the liquid-air interface points inwards, in order to minimize the liquid-air surface area. However, the force due to the solid-air interface points outward, in order to maximize the liquid-solid interface, or minimize the solid-air interface. At equilibrium, the water is spread in a thin sheet on the plane; a configuration in which the horizontal component of both forces cancel.

VI. Conclusion

The Finite Volume Particle Method is an attractive method for the computation of fluid flow problems including complex moving boundaries due to its consistency, conservative and ALE properties. However, directionality and hard interaction of the cubic-supported kernels can badly affect the accuracy and stability of the computations. Employing a spherical-supported top-hat kernel can alleviate these issues but requires a precise and fast algorithm for computing the spherical interaction vectors.

In this paper, we have presented a surface partitioning algorithm that can handle any type of geometries formed on a sphere after being intersected by other spheres. Then, we have proposed a method to exactly evaluate the area vector of the partitions. These values are used for exact computation of the interaction vectors and analytic evaluation of surface area gradients.

We have then presented an innovative surface tension model for particle methods with spherical support. The method uses symmetrical particle-particle interaction but, unlike microscopic cohesion-based models, the method is derived from macroscopic considerations. The inputs to the model are surface tension and equilibrium contact angle, and the force is computed at little cost using the surface area gradients evaluated during the partitioning process. We have shown through simulations of drops in various configurations that the

proposed method properly reproduces the physics of surface tension regardless of the wetting conditions.

ACKNOWLEDGMENT

The authors would like to thank the Swiss Commission for Technology and Innovation (CTI), and General Electric GE Renewable Energy for their financial support and technical assistance to this research within the project GPU-SPHEROS grant no. 17568.1 PFEN-IW.

REFERENCES

[1] D. Hietel, K. Steiner, and J. Struckmeier, "A finite-volume particle method for compressible flows," *Mathematical Models and Methods in Applied Sciences*, vol. 10, no. 9, pp. 1363–1382, 2000.

[2] R. M. Nestor, M. Basa, M. Lastiwka, and N. J. Quinlan, "Extension of the finite volume particle method to viscous flow," *Journal of Computational Physics*, vol. 228, no. 5, pp. 1733–1749, 2009.

[3] E. Jahanbakhsh, C. Vessaz, and F. Avellan, "Finite Volume Particle Method for 3-D Elasto-Plastic Solid Simulation," in *Proceedings of the 9th international SPHERIC workshop, Paris*, 2014.

[4] E. Jahanbakhsh, A. Vessaz, Christian Maertens, and F. Avellan, "Finite Volume Particle Method for Fluid-Structure Interaction," in *Proceedings of the 10th international SPHERIC workshop, Parma*, 2015.

[5] N. J. Quinlan and R. M. Nestor, "Fast exact evaluation of particle interaction vectors in the finite volume particle method," in *Meshfree Methods for Partial Differential Equations V*. Springer Berlin Heidelberg, 2011, pp. 219–234.

[6] C. Vessaz, E. Jahanbakhsh, and F. Avellan, "Flow simulation of a pelton bucket using finite volume particle method," *IOP Conference Series: Earth and Environmental Science*, vol. 22, no. 1, p. 012003, 2014.

[7] C. Vessaz, E. Jahanbakhsh, and F. Avellan, "Flow simulation of jet deviation by rotating pelton buckets using finite volume particle method," *ASME. J. Fluids Eng.*, vol. 137, no. 7, pp. 074 501–074 501, 2015.

[8] E. Jahanbakhsh, "Simulation of silt erosion using particle-based methods," Ph.D. dissertation, École Polytechnique Fédéral de Lausanne, 2014.

[9] R. A. Gingold and J. J. Monaghan, "Smoothed particle hydrodynamics-theory and application to non-spherical stars," *Monthly Notices of the Royal Astronomical Society*, vol. 181, pp. 375–389, 1977.

[10] R. J. LeVeque, *Finite volume methods for hyperbolic problems*. Cambridge university press, 2002, vol. 31.

[11] M. Junk, "Do finite volume methods need a mesh?" in *Meshfree Methods for Partial Differential Equations*, ser. Lecture Notes in Computational Science and Engineering, M. Griebel and M. Schweitzer, Eds. Springer Berlin Heidelberg, 2003, vol. 26, pp. 223–238.

[12] R. M. Nestor and N. J. Quinlan, "Incompressible moving boundary flows with the finite volume particle method," *Computer Methods in Applied Mechanics and Engineering*, vol. 199, no. 3336, pp. 2249 – 2260, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045782510000927

[13] D. Teleaga and J. Struckmeier, "A finite-volume particle method for conservation laws on moving domains," *International Journal for Numerical Methods in Fluids*, vol. 58, no. 9, pp. 945–967, 2008. [Online]. Available: http://dx.doi.org/10.1002/fld.1778

[14] N. J. Quinlan, L. Lobovsk, and R. M. Nestor, "Development of the meshless finite volume particle method with exact and efficient calculation of interparticle area," *Computer Physics Communications*, vol. 185, no. 6, pp. 1554 – 1563, 2014.

[15] E. Jahanbakhsh, C. Vessaz, A. Maertens, and F. Avellan, "Development of a finite volume particle method for 3-D fluid flow simulations," *Computer Methods in Applied Mechanics and Engineering*, vol. 298, pp. 80 – 107, 2016.

[16] N. Akinci, G. Akinci, and M. Teschner, "Versatile Surface Tension and Adhesion for SPH Fluids," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 182:1–182:8, Nov. 2013. [Online]. Available: http://doi.acm.org/10.1145/2508363.2508395

[17] C. Vessaz, E. Jahanbakhsh, and F. Avellan, "FPM simulations of a high-speed water jet validation with CFD and experimental results," in *Advances in Hydroinformatics*, ser. Springer Hydrogeology, P. Gourbesville, J. Cunge, and G. Caignaert, Eds. Springer Singapore, 2014, pp. 419–431.

[18] J. Monaghan, "Simulating free surface flows with SPH," *Journal of Computational Physics*, vol. 110, no. 2, pp. 399 – 406, 1994.

[19] E. Jahanbakhsh, A. Maertens, N. J. Quinlan, C. Vessaz, and F. Avellan, "Exact finite volume particle method with spherical-support kernels," *Computer Methods in Applied Mechanics and Engineering*, 2016, under review.

[20] E. W. Weisstein, "Spherical Cap," from MathWorld–A Wolfram Web Resource. [Online]. Available: http://mathworld.wolfram.com/SphericalCap.html

[21] E. W. Weisstein, "Small Circle," from MathWorld–A Wolfram Web Resource. [Online]. Available: http://mathworld.wolfram.com/SmallCircle.html

[22] E. W. Weisstein, "Great Circle," from MathWorld–A Wolfram Web Resource. [Online]. Available: http://mathworld.wolfram.com/GreatCircle.html

[23] R. G. Chamberlain and W. H. Duquette, "Some algorithms for polygons on a sphere," in *JPL TRS 1992+*. Pasadena, CA : Jet Propulsion Laboratory, National Aeronautics and Space Administration, April 2007.

[24] J. P. Morris, "Simulating surface tension with smoothed particle hydrodynamics," *International Journal for Numerical Methods in Fluids*, vol. 33, no. 3, pp. 333–353, Jun. 2000. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/1097-0363(20000615)33:3¡333::AID-FLD11¿3.0.CO;2-7/abstract

[25] M. Mller, D. Charypar, and M. Gross, "Particle-based fluid simulation for interactive applications," in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2003, pp. 154–159. [Online]. Available: http://dl.acm.org/citation.cfm?id=846298

[26] S. Adami, X. Y. Hu, and N. A. Adams, "A new surface-tension formulation for multi-phase SPH using a reproducing divergence approximation," *Journal of Computational Physics*, vol. 229, no. 13, pp. 5011–5021, Jul. 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999110001324

[27] J. U. Brackbill, D. B. Kothe, and C. Zemach, "A continuum method for modeling surface tension," *Journal of Computational Physics*, vol. 100, no. 2, pp. 335–354, Jun. 1992. [Online]. Available: http://www.sciencedirect.com/science/article/pii/002199919290240Y

[28] T. Breinlinger, P. Polfer, A. Hashibon, and T. Kraft, "Surface tension and wetting effects with smoothed particle hydrodynamics," *Journal of Computational Physics*, vol. 243, pp. 14–27, Jun. 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999113001575

[29] M. Bussmann, J. Mostaghimi, and S. Chandra, "On a three-dimensional volume tracking model of droplet impact," *Physics of Fluids (1994-present)*, vol. 11, no. 6, pp. 1406–1417, Jun. 1999. [Online]. Available: http://scitation.aip.org/content/aip/journal/pof2/11/6/10.1063/1.870005

[30] S. Nugent and H. A. Posch, "Liquid drops and surface tension with smoothed particle applied mechanics," *Physical Review E*, vol. 62, no. 4, pp. 4968–4975, Oct. 2000. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevE.62.4968

[31] A. Tartakovsky and P. Meakin, "Modeling of surface tension and contact angles with smoothed particle hydrodynamics," *Physical Review E*, vol. 72, no. 2, p. 026301, Aug. 2005. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevE.72.026301