6th European Conference on Computational Mechanics (ECCM 6)
7th European Conference on Computational Fluid Dynamics (ECFD 7)
11âĂŞ15 June 2018, Glasgow, UK

A ROBUST PARTICLE–BASED SOLVER FOR MODELING HEAT TRANSFER IN MULTIPHASE FLOWS

M. AFRASIABI[1,2], M. ROETHLIN[2], E. CHATZI[1] AND K. WEGENER[2]

[1] Institute of Structural Engineering (IBK), D-BAUG, ETH Zürich
Stefano-Franscini-Platz 5, 8093 Zürich, Switzerland
E-mail:afrasiabi@ethz.ch
http://www.chatzi.ibk.ethz.ch/people.html

[2] Institute of Machine Tools & Manufacturing (IWF), D-MAVT, ETH Zürich
Leonhardstrasse 21, 8092 Zürich, Switzerland
https://www.iwf.mavt.ethz.ch/people/staff/index

Key words: Multiphase flows; Heat transfer; Moving interfaces; Meshfree methods; Level Set.

Abstract. This article presents a novel coupling of three state-of-the-art particle methods, capable of simulating heat transfer in multiphase flows with minimum error. The idea originates from increasing the robustness of the geometric features required to calculate physical quantities, such as the boundary conditions in the heat equation and the surface tension force in Navier-Stokes equations. The workability of this new approach for handling thermal effects in complex moving interfaces is examined with a numerical test case wherein multiple scenarios such as internal flow, bursting at free surface as well as bubble evolutionary motions occur. Performing a benchmark comparison with the reference model implemented in COMSOL Multiphysics®, a markedly improved mass conservation using the proposed particle-based solver is observed. This, in turn, results in a more realistic and accurate approximation of the temperature.

1 INTRODUCTION

Thermal simulation of multiphase flows requiring accurate prediction of field variables occur in many engineering problems within the scope of Computational Fluid Dynamics (CFD). There is thus a clear need for an accurate CFD solver capable of handling distorted flows and thermal issues simultaneously. Having formed robust solutions for these problems, the original meshfree method Smoothed Particle Hydrodynamics (SPH) [1, 2] is being extensively applied to numerous CFD problems [3, 4, 5, 6, 7, 8] after almost 40 years from its genesis.

Despite significant advances in developing other solutions, the original formulation of SPH for fluid dynamics, known as weakly-compressible SPH (WCSPH), is a well-established technique to handle the incompressibility condition in fluid flows. Albeit at the price of numerical fluctuations in the pressure and velocity fields [4], the WCSPH formulation exploits a proper equation of state to link density with pressure. The density is thereby allowed to vary theoretically within about 1%–5% of the reference value [3]. The successful use of this approach has been shown to exhibit a practical workaround to various problems in the context of multiphase flows (e.g., [9, 10]).

The family of the Finite Element (FE) methods offers an adept tool for simulation of heat transfer problems and has been widely exploited in both the academia and the industry [11]. When compared against meshfree methods, the excellence of FE methods in thermal analysis chiefly stems from their robustness in handling the heat equation (without facing the difficulty of discretizing the Laplacian operator) and the straightforwardness in imposing the boundary conditions, plus a relatively low computational efforts that they offer. Nevertheless, a multiplicity of algorithms can be devised to enhance meshfree methods in this regard, such as the categories of boundary treatments [12], partition of unity by smoothing kernel reconstruction [13, 14], and approximation of higher-order spatial derivatives [15, 16, 17, 18], to name a few.

Another major focus of this research is on the design/adoption of a versatile computational tool for representing the interface evolutions, because most CFD simulations center around generating newly developed surfaces by fluid flows. Apart from the original Level Set (LS) methods [19, 20], the hybrid Particle Level Set (PLS) approaches have been successfully attempted in a broad array of applications [21, 22, 23, 24]. The elegance in PLS methods relies on the coexistence of particles and grids, by which they combine the accuracy of Lagrangian advection with the simplicity of the Eulerian level set surface representation.

In the wake of the aforementioned literature investigation, this paper aims at presenting a new coupling of three meshfree solvers to efficiently handle thermal simulations in multiphase flows. Towards this end, we employ a selective WCSPH scheme for taking care of the multiphase flows, coupled with the most recent yet accurate meshfree scheme for solving the heat equation. This scheme was first introduced by Korzilius et al. in 2016 [18] for static particles, and is herein for the first time extended to dynamic particles. An enhanced version of the PLS technique in [22], hereinafter EPLS, is presented for capturing the moving interfaces in the course of this work. The enhancement is basically due to the utilization of a second-order reinitialization method.

2 `TOOLKIT#1`: CFD SOLVER

A selective WCSPH for multiphase flows is presented. The governing equations are summarized for particle i associated with a velocity field $\underline{v}_i$. In what follows, ρ_i is the density of particle i, μ_i is its dynamic (shear) viscosity and $\underline{g}$ represents the gravitational acceleration. The superscripts "pr", "vi", "st", and "bf" are the abbreviations for the pressure, viscous, surface tension, and body force contributions to the momentum equation, respectively. The smoothing kernel $W_{ij}(\underline{r}_{ij}, h)$ is shortened to W_{ij} wherein h denotes the smoothing length, and $\underline{r}_{ij}$ is the relative distance vector between particle i and j. The gradient of W_{ij} is ∇W_{ij}, and the relative velocity between particle i and j is $\underline{v}_{ij} = \underline{v}_i - \underline{v}_j$. Also, the volume contribution of particle j is represented by $V_j = \frac{m_j}{\rho_j}$ in which m_j denotes the mass of particle j. Note that in this paper $\langle \diamondsuit \rangle$ indicates the numerical approximation of $\diamondsuit$, where the first, second, and forth order tensors are denoted as $\underline{\diamondsuit}$, $\underline{\underline{\diamondsuit}}$, and $\underset{\approx}{\diamondsuit}$, respectively.

$$\begin{cases} \langle \rho_i \rangle = \frac{m_i}{\langle V_i \rangle} = m_i \sum_{j=1}^{N} W_{ij} \\ \rho_i \frac{D\underline{v}_i}{Dt} = \underbrace{-\nabla p_i}_{\underline{\mathrm{F}}_i^{pr}} + \underbrace{\mu_i \nabla^2 \underline{v}_i}_{\underline{\mathrm{F}}_i^{vi}} + \underbrace{\underline{f}_i^{st}}_{\underline{\mathrm{F}}_i^{st}} + \underbrace{\rho_i \underline{g}}_{\underline{\mathrm{F}}_i^{bf}} \\ p_i = p_0 \left[\left(\frac{\rho}{\rho_0} \right)_i^{\gamma} - 1 \right] + \chi \end{cases} \tag{1}$$

in which $p_0 = \frac{c^2 \rho_0}{\gamma}$ and, according to [3], $\gamma = 1.4$ and $\gamma = 7$ are used for gases and fluids, respectively. In the expression of p_0, c stands for the artificial sound speed in the given fluid (usually $c \geq 10 \times |\underline{v}_{max}|$ if $\underline{v}_{max}$ is the expected maximum velocity in the fluid, as discussed in [4]). In this work, a positive background pressure $\chi = 0.05\, p_0$ is utilized to prevent the incidence of negative pressures, i.e., tensile instability mode, in the numerical simulation. The following forms are considered for the symbolic terms of the PDE written in the second line of Eq. (1), as outlined in [4, 7, 10, 25, 26]

$$\begin{cases} \langle \underline{F}_i^{pr} \rangle = -\frac{1}{V_i} \sum_{j=1}^{N} \left(p_i V_i^2 + p_j V_j^2 \right) \nabla W_{ij} \\ \langle \underline{F}_i^{vi} \rangle = \eta_d \sum_{j=1}^{N} V_j \mu_{ij} \frac{\underline{v}_{ij} \cdot \underline{r}_{ij}}{|\underline{r}_{ij}|^2 + (\epsilon h)^2} \nabla W_{ij} \\ \langle \underline{F}_i^{st} \rangle = -\sigma \kappa_i |\nabla c_i| \underline{\hat{n}}_i \\ \underline{F}_i^{bf} = \rho_i \underline{g} \end{cases} \tag{2}$$

The "quintic spline" function is employed as the smoothing kernel for the numerical simulation in this work, named as "the most efficient kernel" among 10 alternatives by Hongbin et al. in [27].

2.1 Wall Boundary Condition

To enforce the boundary conditions in multiphase SPH models, the generalized wall approach, first introduced by Adami et al. [10], is employed in this work by the virtue of the "*ghost particles*". A full explanation of this approach can be found in [10].

2.2 Time Integration

A second-order velocity–Verlet scheme is adopted in order to update the equation of motion and the momentum equation in time [28]. This scheme is frequently used by the CFD community, as outlined in [25, 29]. The step size of this time integration, Δt, is calculated such that it meets the following criterion globally.

$$\Delta t = 0.5 \times \min \left(\Delta t_{vi}, \Delta t_{cs}, \Delta t_{bf}, \Delta t_{st} \right) \tag{3}$$

wherein the CFL-conditions for the viscous, artificial sound speed, body force, and surface tension, are respectively set to $CFL_{vi} = 0.125$, $CFL_{cs} = 0.25$, $CFL_{bf} = 0.25$, and $CFL_{st} = 0.25$.

3 TOOLKIT#2: THERMAL SOLVER

In this section, the Improved Corrective Smoothed Particle Methods (ICSPM) formulation for solving the heat equation is derived in a mathematically non-rigorous manner. The general governing PDE for transient heat transfer is often expressed as

$$\rho c_p \frac{\partial T}{\partial t} = \nabla(k\nabla T) + Q_{in} - Q_v \tag{4}$$

where c_p denotes specific heat capacity, T temperature, k heat conductivity, Q_{in} the heat source and Q_v heat loss per time and volume. The surface heat source, Q_{in}, originates from the internal flows and will be described in the respective example in Sec. 5. Q_v is herein assumed to be zero, meaning no heat is lost upon radiation, nor via convection. In this work, the ICSPM is exploited for discretizing the heat conduction equation, i.e., (4). This leads to a fully second-order approximation of the heat equation, benefiting from a very good compromise of the computational cost vs. accuracy.

Initially suggested by Korzilius et al. and used for fixed-in-space particles in [18], the ICSPM approximation for the second derivatives and Laplacian operator of T are calculated from

$$\Big\langle \nabla\nabla T_i \Big\rangle = \underset{\approx}{\boldsymbol{D}}_i^{-1} \left[\sum_{j=1}^{N} V_j (T_j - T_i) \nabla\nabla W_{ij} - \sum_{j=1}^{N} V_j \, \underline{r}_{ji} \cdot \underline{\underline{A}}_i^{-1} \sum_{j=1}^{N} V_j (T_j - T_i) \nabla W_{ij} \; \nabla\nabla W_{ij} \right] \tag{5}$$

$$\Big\langle \Delta T_i \Big\rangle = tr\left[\Big\langle \nabla\nabla T_i \Big\rangle \right] \tag{6}$$

where, for 2D problems, the ICPSM renormalization tensor $\underset{\approx}{\boldsymbol{B}}_i$ for particle i is defined as

$$\underset{\approx}{\boldsymbol{D}}_i = \underset{\approx}{\boldsymbol{B}}_i - \left[\left(\sum_{j=1}^{N} V_j (\underline{r}_{ji} \otimes \nabla\nabla W_{ij}) \right) \cdot \underline{\underline{A}}_i \cdot \left(\sum_{j=1}^{N} V_j (\underline{\underline{C}}_{ij} \otimes \nabla W_{ij}) \right) \right] \tag{7}$$

with

$$\underline{\underline{A}}_i = \sum_{j=1}^{N} V_j \left(\underline{r}_{ji} \otimes \nabla W_{ij} \right) \tag{8}$$

$$\underset{\approx}{\boldsymbol{B}}_i = \sum_{j=1}^{N} V_j \left(\underline{\underline{C}}_{ij} \otimes \nabla\nabla W_{ij} \right) \tag{9}$$

$$\underline{\underline{C}}_{ij} = \begin{bmatrix} \frac{1}{2}(x_i - x_j)(x_i - x_j) & (x_i - x_j)(y_i - y_j) \\ (y_i - y_j)(x_i - x_j) & \frac{1}{2}(y_i - y_j)(y_i - y_j) \end{bmatrix} \tag{10}$$

if $\otimes$ indicates the tensor product. The final form of the second-order approximation of the Laplacian of the temperature T (or any other field function of interest) at particle i using ICSPM is the sum of the elements on the main diagonal in Eq. (6).

4 TOOLKIT#3: GEOMETRIC SOLVER

This section is not intended to introduce the PLS, reinitialization and remeshing methods from the ground-up, but to detail the design of the proposed particle-based method for representing the geometry. The EPLS methodology in this work is fundamentally built on the Lagrangian representation of LS methods as elaborated upon in [22, 23, 24], and is only projected here for the sake of consistency. The algorithm initiates with the configuration of particles and then performs a consistent remeshing procedure [30, 31, 32] onto uniformly-spaced particles to regularize the particle locations when the particle map gets distorted by the advection field. One starting point for the derivation of the EPLS is the well-known advection equation. Given a body in the Lagrangian framework, the evolution of this body is simply comprehended by solving a system of Ordinary Differential Equations (ODEs) on the representative points (i.e., particles) as follows

$$\begin{aligned} \frac{D\varphi_p}{Dt} &= 0 \\ \frac{D\underline{x}_p}{Dt} &= \underline{v}_p \\ \frac{DV_p}{Dt} &= V_p \left(\nabla \cdot \underline{v}_p \right) \end{aligned} \tag{11}$$

in which $\underline{x}_p$ and V_p are the position vector and volume of particle p, respectively, and φ_p is its level set value. In order to design a truly second-order PLS for a precise representation of the interface, one needs to ascertain a second/higher-order accuracy of each subset. The four main constitutive subsets of the proposed solver are: the "particle approximation" for field variables and their spatial derivatives, the "time integration" scheme for particle advection, the "interpolation kernel" for the remeshing procedure, and the accurate "reinitialization" method. The term "second-order accurate" is related to the order of convergence with increase of N discretization points, and should be distinguished from the "second-order completeness", which indicates the reconstruction of second-order polynomials with null error.

4.1 Particle Approximations

According to Eq. (11), φ_p is kept unchanged on particles during the evolution. This implies that the level set values of the particles are calculated for the problem at hand

only once. As already discussed in Sec. 3, a proper second-order meshfree scheme can be utilized for constructing the first two derivatives in this paper. The particle approximation is required for the computation of the geometric entities such as the unit normal and the curvature of the interface. The unit normal $\underline{n}$ and curvature κ at non-physical particle p are then given by

$$\underline{n}_p = \frac{\nabla \varphi_p}{|\varphi_p|} \tag{12}$$

$$\kappa_p = \nabla \cdot \frac{\nabla \varphi_p}{|\varphi_p|} \tag{13}$$

The ICSPM approximations previously derived in Sec. 3 can be substituted for the gradient and Laplacian operators in the formula above.

4.2 Particle Advection

A fourth order Runge–Kutta (RK4) method is used for evolving the non-physical particles. This choice would guarantee the satisfaction of the second condition needed for the proposed EPLS. While the particles' distance to the interface remains constant within the evolution, their volumes and positions are updated in accordance with Eq. (11).

4.3 Remeshing

The remeshing algorithm is indeed the backbone of the EPLS approach, and originally used by Hieber et al. [22] in the context of PLS methods. From left to right in Fig. 1, the schematic depiction of this procedure is illustrated step by step, which follows the same structure presented in [33].

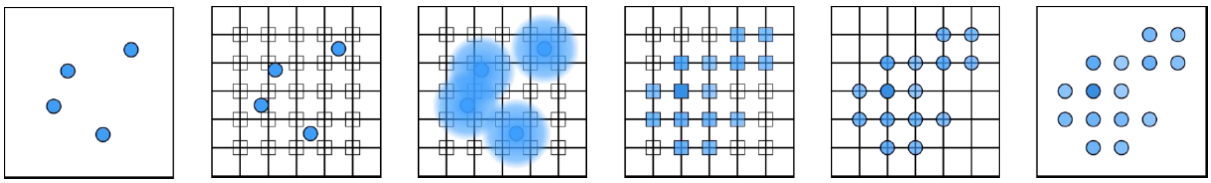

Figure 1: Illustration of the remeshing procedure: (a) scattered particles are given, (b) a superimposed regular mesh is generated, (c) the interpolation kernel on each particle is performed, (d) affected regular grids are found, (e) the affected regular grids become new particles, (f) non-affected mesh grids are deleted

For interpolation between grid nodes and particles, we make use of the commonly-used M_4' kernel, as expressed in [22]

$$M_4'(s) = \begin{cases} 1 - \frac{1}{2}(5s^2 - 3s^3)^2 & \text{if } 0 \leqslant s < 1 \\ \frac{1}{2}(2-s)^2(1-s) & \text{if } 1 \leqslant s \leqslant 2 \\ 0 & \text{if } 2 < s \end{cases} \tag{14}$$

which interpolates the characteristic s by preserving the moment conservation up to the 2^{nd} order, and creates a smooth interpolated function which is mathematically $\mathbb{C}^1$ continuous. For 2D geometries, the interpolation kernel is the Cartesian tensorial multiplicative of its 1D counterparts in x and y directions. The frequency of the remeshing procedure is based on a measure of distortion, which was defined in [22], and is used in this work by setting the distortion threshold to 10^{-6}.

4.4 Reinitialization

Considering the Signed Distance Function (SDF) as the initial values for φ, this does not necessarily preserve its property after evolving under a general velocity field. A remedy to this shortcoming is to re-initialize the LS values and force them to hold their initial SDF property throughout the simulation (see [20]). There are several widely-used reinitialization alternatives in the literature, among which the Fast Marching Method (FMM) [34] has seen a wider range of applications. The "Improved FMM" reinitialization scheme employed in this work, termed as IFMM for the sake of brevity, is an improved Higher Accuracy Fast Marching Method (HAFMM) which was introduced by [35], and further adopted in the work of Hieber et al. [22]. Two types of enhancements are here applied to the current HAFMM, enabling a second-order accuracy of the reinitialization method. Firstly, the ignored information provided by diagonal neighboring particles in Cartesian coordinates are retrieved by incorporating the multistencils framework as suggested by Hassouna et al. [36]. Secondly, a bicubic interpolation scheme is adopted for initializing the fast marching method. This is achieved by constructing a second-order approximation of the interface generated from local data on the particles. It was initially shown by Chopp in [37] that this improvement allows for a second-order approximation of the distance to the interface which can then be used to produce second-order accurate initial conditions for FMM.

5 RESULTS AND DISCUSSION

A buoyancy-driven oil bubble is investigated here to assess the performance of the proposed particle-based software in heat transfer problems, with the parameters listed in Tab. 1. Two different sets of particles with an equal particle spacing, namely "*physical*" and "*geometric*", are separately introduced to carry the associated physical and geometric characteristics. As the velocity field is not known *a priori* in this setting, the velocity of the physical particles are interpolated into the locations of the geometric particles using the quintic spline kernel. Three layers of ghost particles are used for mimicking the wall boundaries, ensuring the completeness of the supporting domain for the chosen smoothing kernel (i.e., the quintic spline kernel with cut-off radius of $3h$). A relatively low resolution of 41×61 particles is considered to highlight the robustness of the proposed solver even without using fine discretization. The moving fluid-fluid interface due to the buoyancy effect is captured by invoking the velocity field computed from `TOOLKIT#1`. The physical scenario in this immersed oil bubble is vividly observed through the consecutive snapshots in Fig 3. As expected, the rising bubble merges with the oil phase, oscillates transiently, reaches to the stability and stands still eventually.

Table 1: Summary of the parameters used

Property	*Dimension*	*Value*
Oil density	kg m^{-3}	879
Water density	kg m^{-3}	1000
Oil viscosity	Pa s	0.0208
Water viscosity	Pa s	0.00101
Surface tension coefficient	N/m	0.015
Oil specific heat capacity	J kg^{-1} K^{-1}	1970
Water specific heat capacity	J kg^{-1} K^{-1}	4185
Oil heat conductivity	W m^{-1} K^{-1}	0.150
Water heat conductivity	W m^{-1} K^{-1}	0.585
Oil initial temperature	K	343
Water initial temperature	K	293
Wall initial temperature	K	318
Width of container	m	0.010
Height of container	m	0.015
Initial position of bubble	m	$(0.0, 0.004)$
Initial diameter of bubble	m	0.004

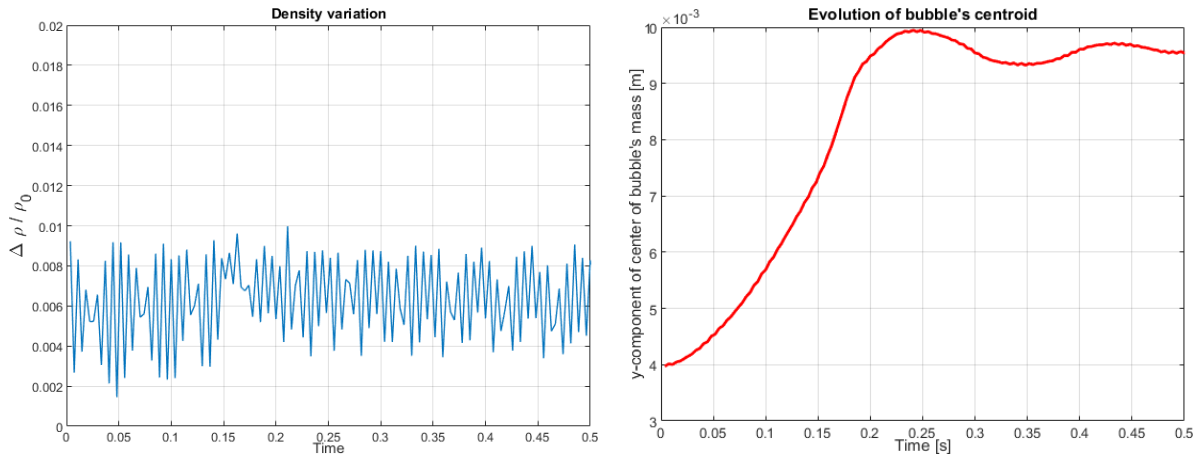

Figure 2: Variation of the maximum density error (left) and evolution of the y–component of the bubble's center of mass (right)

As plotted in Fig. 2, one can notice that the maximum variation in density is reasonably less than the limit of $1\% - 5\%$ required for WCSPH credibility. The y–component of the bubble's centroid is also tracked as a measure of the bubble evolution in time, displayed in Fig. 2. The EPLS result shows a good qualitative agreement with the corresponding output from COMSOL, as shown in Fig. 3. The FE-based LS solution in COMSOL is obtained by setting the CFD modules as predefined physics interfaces, augmented by the *"Laminar Two-Phase Flow"* together with the *"Level Set"* interfaces. One of the salient added values of the proposed meshfree software is that it delivers acceptable accuracy even with such a low resolution, a fascinating attribute from the computational cost point of view.

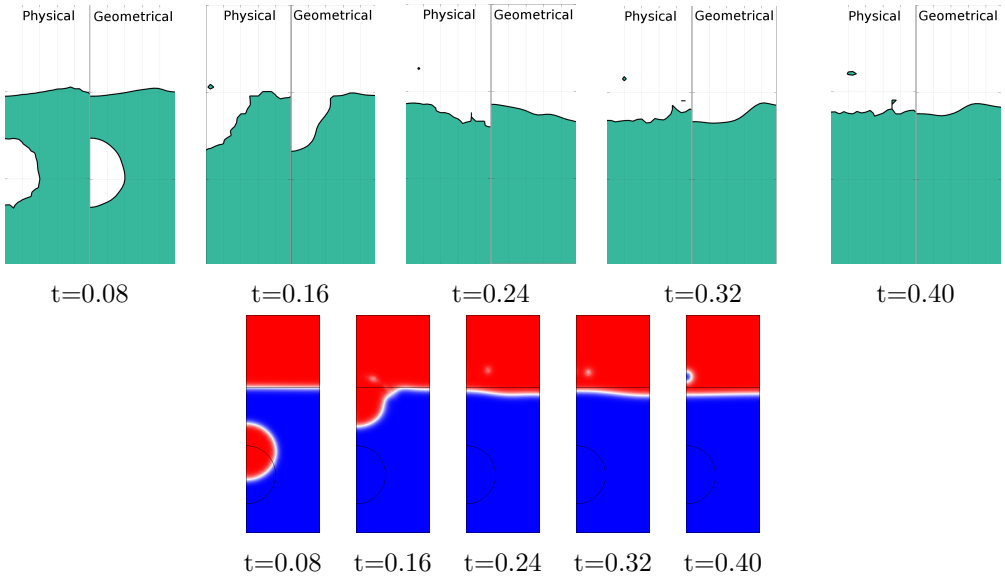

Figure 3: Interface snapshots using EPLS (top) and COMSOL Multiphysics® (bottom)

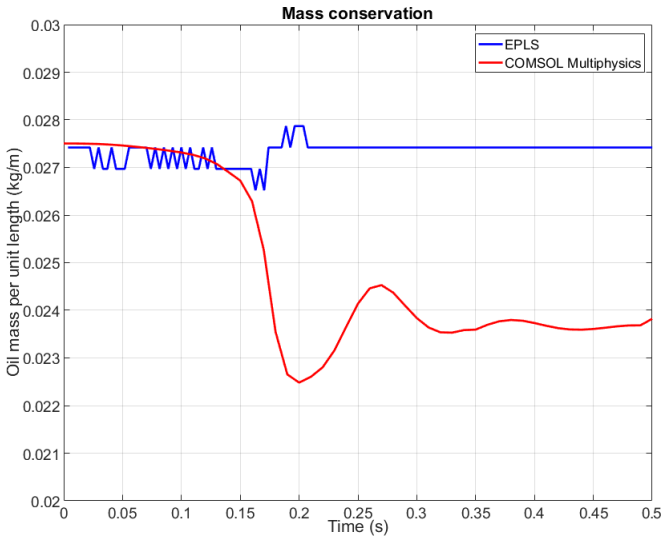

Figure 4: Quantitative comparison of mass conservation

Fig. 4 permits to observe a better conservation of mass using the EPLS solver. What is more, the EPLS solution acts indifferent to the two critical moments of $t \simeq 0.20$ and $t \simeq 0.26$, whereas the mass conservation in COMSOL is greatly influenced. These moments are associated with the instants when the uprising oil bubble first hits the stagnant oil layer above and creates several transient waves before reaching the steady state. The infinitesimal mass gain in EPLS at $t \simeq 0.2$ that can be noticed in Fig. 4 comes from the

remeshing with regards to the impulsive distortion at that time.

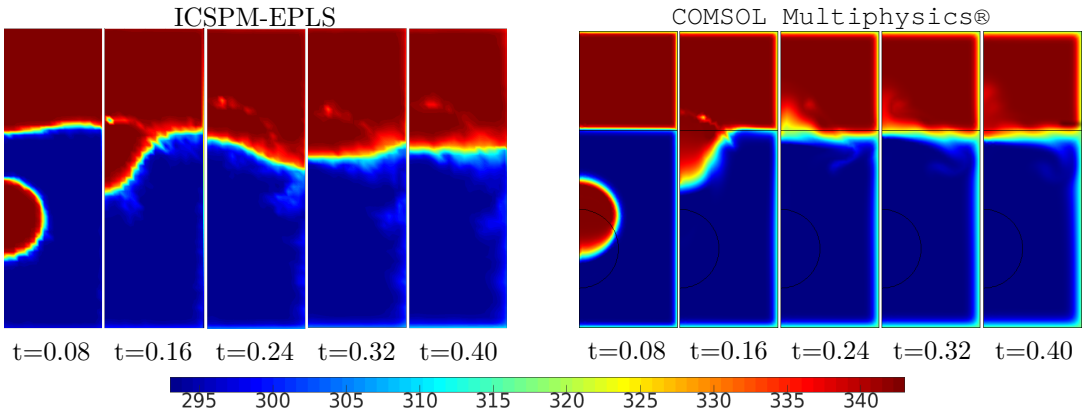

Figure 5: Comparison of temperature distributions in five consecutive times

Finally, the ICSPM scheme is coupled with the EPLS method for addressing the thermal effect in this model. Depicted in Fig. 5 is the temperature distribution using both the present particle-based solver and the solution of COMSOL for the five corresponding time instants. This comparison conveys a good agreement between the two results.

6 CONCLUSIONS

- A new particle-based software is developed and used for the simulation of the heat transfer in a CFD problem. The novelty of this meshfree coupling stems from the robustness of the three interactive solvers, namely the selective WCSPH for multiphase flows (TOOLKIT#1), the ICSPM for solving the heat equation (TOOLKIT#2), and the EPLS scheme for capturing moving interfaces (TOOLKIT#3). Using the proposed numerical tool, a potentially second-order approximation to the overall solution is anticipated. As demonstrated in the rising oil-bubble example, this enables a significantly better conservation of mass compared to FE results, thus a more accurate and reliable approximation of the temperature. Moreover, a major improvement is gained by increasing the accuracy of the geometric features required for computing the physical quantities, e.g., the boundary conditions in the thermal part. Adopting the proposed software would offer yet another added value: allowing for stable and precise CFD simulations with fairly low number of particles. The present particle-based coupling is shown to achieve efficiency previously impossible for the same resolution, creating a vast potential for the simulation of many industrial CFD engineering applications where the accuracy is a major concern.

- Whilst this research provides a substantial step for thermal modeling of CFD problems, several areas of development are still needed. Future work shall contain the extension of the proposed solver to 3D. This might as well encounter the parallel and distributed computing implementations on both CPUs and GPUs. Although being a

daunting task to implement, consideration of a "narrow-band" approach also needs to be evaluated. That is, the geometric particles in "TOOLKIT#3" are seeded only into a limited narrow-band around the interface rather than the entire domain. By doing so, the computational complexity of the EPLS algorithm could be drastically reduced. Further investigation of the method used for a variety of cases involving ill-defined wall boundaries should be made, too. The prescribed velocity and acceleration at ghost particles would no longer be zero in such cases.

References

[1] Leon B Lucy. A numerical approach to the testing of the fission hypothesis. *The astronomical journal*, 82:1013–1024, 1977.

[2] Robert A Gingold and Joseph J Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly notices of the royal astronomical society*, 181(3):375–389, 1977.

[3] Joe J Monaghan. Simulating free surface flows with sph. *Journal of computational physics*, 110(2):399–406, 1994.

[4] Joseph P Morris, Patrick J Fox, and Yi Zhu. Modeling low reynolds number incompressible flows using sph. *Journal of computational physics*, 136(1):214–226, 1997.

[5] M Afrasiabi and S Mohammadi. Analysis of bubble pulsations of underwater explosions by the smoothed particle hydrodynamics method. In *ECCOMAS International Conference on Particle Based Methods, Spain*, 2009.

[6] Nicolas Grenier, Matteo Antuono, Andrea Colagrossi, David Le Touzé, and B Alessandrini. An hamiltonian interface sph formulation for multi-fluid and free surface flows. *Journal of Computational Physics*, 228(22):8380–8393, 2009.

[7] Aman Zhang, Pengnan Sun, and Furen Ming. An sph modeling of bubble rising and coalescing in three dimensions. *Computer Methods in Applied Mechanics and Engineering*, 294:189–209, 2015.

[8] SJ Lind, PK Stansby, and Benedict D Rogers. Incompressible–compressible flows with a transient discontinuous interface using smoothed particle hydrodynamics (sph). *Journal of Computational Physics*, 309:129–147, 2016.

[9] XY Hu and Nikolaus A Adams. A multi-phase sph method for macroscopic and mesoscopic flows. *Journal of Computational Physics*, 213(2):844–861, 2006.

[10] S Adami, XY Hu, and NA Adams. A generalized wall boundary condition for smoothed particle hydrodynamics. *Journal of Computational Physics*, 231(21):7057–7075, 2012.

[11] Georgios Pastras, Apostolos Fysikopoulos, Panagiotis Stavropoulos, and George Chryssolouris. An approach to modelling evaporation pulsed laser drilling and its energy efficiency. *The International Journal of Advanced Manufacturing Technology*, 72(9-12):1227–1241, 2014.

[12] PW Randles and LD Libersky. Smoothed particle hydrodynamics: some recent improvements and applications. *Computer methods in applied mechanics and engineering*, 139(1):375–408, 1996.

[13] Wing Kam Liu, Sukky Jun, and Yi Fei Zhang. Reproducing kernel particle methods. *International journal for numerical methods in fluids*, 20(8-9):1081–1106, 1995.

[14] Gordon R Johnson and Stephen R Beissel. Normalized smoothing functions for sph impact computations. *International Journal for Numerical Methods in Engineering*, 39(16):2725–2741, 1996.

[15] JK Chen, JE Beraun, and TC Carney. A corrective smoothed particle method for boundary value problems in heat conduction. *International Journal for Numerical Methods in Engineering*, 46(2):231–252, 1999.

[16] Jeff D Eldredge, Anthony Leonard, and Tim Colonius. A general deterministic treatment of derivatives in particle methods. *Journal of Computational Physics*, 180(2):686–709, 2002.

[17] R Fatehi and MT Manzari. Error estimation in smoothed particle hydrodynamics and a new scheme for second derivatives. *Computers & Mathematics with Applications*, 61(2):482–498, 2011.

[18] SP Korzilius, WHA Schilders, and MJH Anthonissen. An improved cspm approach for accurate second-derivative approximations with sph. *Journal of Applied Mathematics and Physics*, 5(01):168, 2016.

[19] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988.

[20] Stanley Osher and Ronald Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153. Springer Science & Business Media, 2006.

[21] Douglas Enright, Ronald Fedkiw, Joel Ferziger, and Ian Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational physics*, 183(1):83–116, 2002.

[22] Simone E Hieber and Petros Koumoutsakos. A lagrangian particle level set method. *Journal of Computational Physics*, 210(1):342–367, 2005.

[23] Simone E Hieber and Petros Koumoutsakos. A lagrangian particle method for the simulation of linear and nonlinear elastic models of soft tissue. *Journal of Computational Physics*, 227(21):9195–9215, 2008.

[24] G-H Cottet, J-M Etancelin, Franck Pérignon, and Christophe Picard. High order semi-lagrangian particle methods for transport equations: numerical analysis and implementation issues. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48(4):1029–1060, 2014.

[25] Joe J Monaghan. Smoothed particle hydrodynamics. *Reports on progress in physics*, 68(8):1703, 2005.

[26] S Adami, XY Hu, and NA Adams. A new surface-tension formulation for multi-phase sph using a reproducing divergence approximation. *Journal of Computational Physics*, 229(13):5011–5021, 2010.

[27] Jin Hongbin and Ding Xin. On criterions for smoothed particle hydrodynamics kernels in stable field. *Journal of Computational Physics*, 202(2):699–709, 2005.

[28] Loup Verlet. Computer" experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Physical review*, 159(1):98, 1967.

[29] S Adami, XY Hu, and Nikolaus A Adams. A transport-velocity formulation for smoothed particle hydrodynamics. *Journal of Computational Physics*, 241:292–307, 2013.

[30] Georges-Henri Cottet and Petros D Koumoutsakos. *Vortex methods: theory and practice*. Cambridge university press, 2000.

[31] AK Chaniotis, D Poulikakos, and P Koumoutsakos. Remeshed smoothed particle hydrodynamics for the simulation of viscous and heat conducting flows. *Journal of Computational Physics*, 182(1):67–90, 2002.

[32] Anas Obeidat and Stéphane Bordas. Three-dimensional remeshed smoothed particle hydrodynamics for the simulation of isotropic turbulence. *International Journal for Numerical Methods in Fluids*, 2017.

[33] Petros Koumoutsakos, Georges-Henri Cottet, and Diego Rossinelli. Flow simulations using particles-bridging computer graphics and cfd. In *SIGGRAPH 2008-35th International Conference on Computer Graphics and Interactive Techniques*, pages 1–73. ACM, 2008.

[34] James A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[35] James Albert Sethian. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, volume 3. Cambridge university press, 1999.

[36] M Sabry Hassouna and Aly A Farag. Multistencils fast marching methods: A highly accurate solution to the eikonal equation on cartesian domains. *IEEE transactions on pattern analysis and machine intelligence*, 29(9):1563–1574, 2007.

[37] David L Chopp. Some improvements of the fast marching method. *SIAM Journal on Scientific Computing*, 23(1):230–244, 2001.