Computer Physics Communications 250 (2020) 107157

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

CPC 50th anniversary article

SPHERA v.9.0.0: A Computational Fluid Dynamics research code, based on the Smoothed Particle Hydrodynamics mesh-less method☆,☆☆

Andrea Amicarelli [a,*], Sauro Manenti [b], Raffaele Albano [c], Giordano Agate [a], Marco Paggi [d], Laura Longoni [e], Domenica Mirauda [c], Latifa Ziane [f], Giacomo Viccione [g], Sara Todeschini [b], Aurelia Sole [c], Lara Martina Baldini [e,a], Davide Brambilla [e], Monica Papini [e], Mohamed Cherif Khellaf [f], Bonaventura Tagliafierro [g], Luca Sarno [g], Guido Pirovano [a]

[a] Ricerca sul Sistema Energetico - RSE SpA, Department SFE, via Rubattino, 54, 20134, Milan, Italy
[b] Dipartimento di Ingegneria Civile e Architettura (DICAr), Università degli studi di Pavia, via Ferrata 3, 27100 Pavia, Italy
[c] School of Engineering, University of Basilicata, Potenza, Italy
[d] IMT School for Advanced Studies Lucca, Piazza San Francesco 19, 55100 Lucca, Italy
[e] Dipartimento di Ingegneria Civile, Ambientale e Territoriale, Politecnico di Milano, Milan, 20133, Italy
[f] LEGHYD laboratory, faculty of civil engineering, University of Sciences and Technology Houari Boumediene, Algiers, Algeria
[g] Department of Civil Engineering, University of Salerno, via Giovanni Paolo II, 132, 84084 Fisciano, Italy

ARTICLE INFO

Article history:
Received 31 May 2019
Received in revised form 19 December 2019
Accepted 3 January 2020
Available online 27 January 2020

Keywords:
SPH
Floods
Landslides
Transport of solid bodies
Bed-load transport
Dense granular flows
SPHERA
Hydrodynamic lubrication

ABSTRACT

SPHERA v.9.0.0 (RSE SpA) is a FOSS CFD-SPH research code validated on the following application fields: floods with transport of solid bodies and bed-load transport; fast landslides and their interactions with water reservoirs; sediment removal from water bodies; fuel sloshing tanks; hydrodynamic lubrication for energy efficiency actions in the industrial sector. SPHERA is featured by several numerical schemes dealing with: transport of solid bodies in fluid flows; treatment of fixed and mobile solid boundaries; dense granular flows and an erosion criterion. The source and executable codes, the input files and the free numerical chain of SPHERA v.9.0.0 are presented. Some reference validations and applications are also provided. SPHERA is developed and distributed on a GitHub public repository.

Program summary

Program title: SPHERA v.9.0.0
Program files doi: http://dx.doi.org/10.17632/pwv9rsf3w8.1
Code Ocean capsule: https://doi.org/10.24433/CO.7457751.v1
Licensing provisions: GNU General Public License 3
Programming language: Fortran 95
Supplementary material: software documentation/guide, 34 tutorials
Nature of problem: SPHERA v.9.0.0 has been applied to free-surface and multi-phase flows involving the following application fields: floods (with transport of solid bodies, bed-load transport and a domain spatial coverage up to some hundreds of squared kilometres), fast landslides and wave motion, sediment removal from water reservoirs, fuel sloshing tanks, hydrodynamic lubrication.
Solution method: SPHERA v.9.0.0 is a research FOSS ("Free/Libre and Open-Source Software") code based on the SPH ("Smoothed Particle Hydrodynamics") technique, a mesh-less Computational Fluid Dynamics numerical method for free surface and multi-phase flows. The five numerical schemes featuring SPHERA v.9.0.0 deal with: dense granular flows; transport of solid bodies in free surface flows; boundary treatment for both mobile and fixed frontiers; 2D erosion criterion.

☆ The review of this paper was arranged by Prof. N.S. Scott.

☆☆ This paper and its associated computer program are available via the Computer Physics Communication homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

* Corresponding author.
E-mail addresses: andrea.amicarelli@rse-web.it (A. Amicarelli), sauro.manenti@unipv.it (S. Manenti), raffaele.albano@unibas.it (R. Albano), giordano.agate@rse-web.it (G. Agate), marco.paggi@imtlucca.it (M. Paggi), laura.longoni@polimi.it (L. Longoni), domenica.mirauda@unibas.it (D. Mirauda), latifa.ziane@hotmail.fr (L. Ziane), gviccion@unisa.it (G. Viccione), sara.todeschini@unipv.it (S. Todeschini), aurelia.sole@unibas.it (A. Sole), lara_baldini@virgilio.it (L.M. Baldini), davide.brambilla@polimi.it (D. Brambilla), monica.papini@polimi.it (M. Papini), mckhellaf@yahoo.fr (M.C. Khellaf), btagliafierro@unisa.it (B. Tagliafierro), lsarno@unisa.it (L. Sarno), guido.pirovano@rse-web.it (G. Pirovano).

https://doi.org/10.1016/j.cpc.2020.107157

Additional comments including restrictions and unusual features: SPHERA v.9.0.0 is a 3D research FOSS ("Free/Libre and Open-Source Software") code (developed under the subversion control system Git) with peculiar features for: floods (with transport of solid bodies, bed-load transport and a domain spatial coverage up to some hundreds of squared kilometres), fast landslides and wave motion, sediment removal from water reservoirs, fuel sloshing tanks, hydrodynamic lubrication. The whole numerical chain of SPHERA is made of FOSS, freeware and Open Data numerical tools.

References:
SPHERA (RSE SpA), https://github.com/AndreaAmicarelliRSE/SPHERA, last access on 28May2019
Amicarelli A., G. Agate, R. Guandalini; 2013; A 3D Fully Lagrangian Smoothed Particle Hydrodynamics model with both volume and surface discrete elements; International Journal for Numerical Methods in Engineering, 95: 419–450, DOI: 10.1002/nme.4514
Amicarelli A., R. Albano, D. Mirauda, G. Agate, A. Sole, R. Guandalini; 2015; A Smoothed Particle Hydrodynamics model for 3D solid body transport in free surface flows; Computers & Fluids, 116:205–228. DOI 10.1016/j.compfluid.2015.04.018
Amicarelli A., B. Kocak, S. Sibilla, J. Grabe; 2017; A 3D Smoothed Particle Hydrodynamics model for erosional dam-break floods; International Journal of Computational Fluid Dynamics, 31(10):413-434; DOI 10.1080/10618562.2017.1422731
Manenti S., S. Sibilla, M. Gallati, G. Agate, R. Guandalini; 2012; SPH Simulation of Sediment Flushing Induced by a Rapid Water Flow; Journal of Hydraulic Engineering ASCE 138(3): 227–311.
Di Monaco A., S. Manenti, M. Gallati, S. Sibilla, G. Agate, R. Guandalini; 2011; SPH modelling of solid boundaries through a semi-analytic approach. Engineering Applications of Computational Fluid Mechanics, 5(1):1-15.

1. Introduction

SPHERA v.9.0.0 (RSE SpA, 2018, [1]) is a research FOSS ("Free/Libre and Open-Source Software", Free Software Foundation, [2]) code based on the SPH ("Smoothed Particle Hydrodynamics") technique, a mesh-less Computational Fluid Dynamics method for free surface and multi-phase flows. SPHERA has been developed to numerically simulate 3D non-stationary fluid dynamics phenomena with complex boundaries for the following application fields: floods (with transport of solid bodies, bed-load transport and a domain spatial coverage up to some hundreds of squared kilometres); fast landslides and wave motion, hydroelectric plants (included the sediment removal from water reservoirs); sloshing tanks; hydrodynamic lubrication for energy efficiency actions in the industrial sector.

The main numerical developments featuring SPHERA v.9.0.0 are listed hereafter:

- scheme for dense granular flows (Amicarelli et al. 2017, [3]);
- scheme for the transport of solid bodies in free surface flows (Amicarelli et al. 2015, [4]);
- scheme for a boundary treatment ("DB-SPH" for simplicity of notation) based on discrete surface and volume elements, and on a 1D Linearized Partial Riemann Solver coupled with a MUSCL (Monotonic Upstream-Centred Scheme for Conservation Laws) spatial reconstruction scheme (Amicarelli et al. 2013, [5]);
- scheme for a 2D erosion criterion (Manenti et al. 2012, [6]);
- scheme for a boundary treatment ("semi-analytic approach or SA-SPH" for simplicity of notation) based on volume integrals, numerically computed outside of the fluid domain (Di Monaco et al. 2011, [7]).

Other major numerical developments are available in SPHERA v.9.0.0 in a preliminary form and deal with the following topics: 2-interface 3D erosion criterion; 3D rotations of solid bodies based on Rodrigues formula; sliding friction force; body–boundary normal reaction forces under sliding; soil liquefaction; shear-stress boundary terms for the DB-SPH scheme; damage scheme for electrical substations; pressure limiters for fluid–structure interactions; bottom drag exerted on the fluid; higher order Runge–Kutta time integration schemes; shear stress gradient terms for laminar flows and fluid–structure interactions under no-slip conditions.

SPHERA is based on the SPH technique, whose features are briefly recalled in Section 1.1.

The paper is structured as follows. The main numerical schemes of SPHERA are synthesized. They deal with: transport of solid bodies and the semi-analytic approach as a boundary treatment scheme for fixed boundaries (Section 2); dense granular flows (Section 3); DB-SPH boundary treatment scheme for both fixed and mobile boundaries (Section 4); time integration (Section 5). Further, the source code, the executable code and the input files of SPHERA are presented (Section 6). All the numerical tools of the modelling chain of SPHERA are free (Section 7). The code tutorials, validations and applications are summarized (Sections 8 and 9). Overall conclusions are finally reported (Section 10).

1.1. The SPH method

Smoothed Particle Hydrodynamics (SPH) is a mesh-less CFD method, whose computational nodes are represented by numerical fluid particles. In the continuum, the functions and derivatives in the fluid dynamics balance equations are approximated by convolution integrals, which are weighted by interpolating (or smoothing) functions, called kernel functions.

The integral SPH approximation ($<>_I$) of a generic function (f) is defined as:

$$\langle f \rangle_{I,\underline{x}_0} = \int_{V_h} fW\left(\underline{x} - \underline{x}_0, h\right) dx^3 \tag{1.1}$$

where W (m^{-3}) is the kernel function (Monaghan, 2005, [8]), $\underline{x}_0$ (m) is the position of a generic computational point and V_h (m^3) is the integration volume, which is called kernel support. This is represented by a sphere of radius $2h$ (m), where h is a characteristic length of the kernel support, which is possibly truncated by the frontiers of the fluid domain.

Any first derivative of a generic function, calculated along i-axis, can be computed as in (1.1), after replacing f with the targeted derivative. After integration by parts, one obtains:

$$\left\langle \frac{\partial f}{\partial x_i} \right| \Bigg\rangle_{I,\underline{x}_0} = \int_{A_h} fWn_i dx^2 - \int_{V_h} f \frac{\partial W}{\partial x_i} dx^3 \tag{1.2}$$

The integration also involves the surface A_h (m^2) of the kernel support whose local orientation is defined by the normal $\underline{n}$. The associated surface integral is non-zero in case of a truncated kernel support. The representation of this term noticeably differentiates the various SPH codes developed by the research community (Adami et al. 2012, [9]; Hashemi et al. 2012, [10]; Macia et al. 2012, [11]; Mayrhofer et al. 2013, [12]; Ferrand et al. 2013, [13]; Amicarelli et al. 2013, [5]).

Far from boundaries, the SPH particle approximation of (1.2) reads:

$$\left\langle \frac{\partial f}{\partial x_i}\middle|_{\underline{x}_0} \right\rangle = -\sum_b f_b \left.\frac{\partial W}{\partial x_i}\right|_b \omega_b \tag{1.3}$$

where a summation on particle volumes ω (m^3) replaces the volume integral. The subscripts "$_0$" and "$_b$" refer to the computational particle and its "neighbouring particles" (fluid particles within the kernel support of the computational particle), respectively. Each particle represents a mobile fluid volume featured by the fluid dynamics physical quantities and a time-dependent position. In order to minimize the SPH truncation errors, initial conditions define any particle volume as a cube and any particle position as coincident with the associated particle volume barycentre. There is no need to define the time evolution of the edges and the barycentre of the SPH particle volumes. More details on the SPH particle discretization are available in Monaghan (2005, [8]).

Usually, the approximation (1.3) is replaced by more complicated and accurate formulas. Further, the SPH method can also approximate a generic nth derivative, analogously to (1.3).

Among the various numerical methods, Smoothed Particle Hydrodynamics (SPH) has several advantages: a direct estimation of free surface and phase/fluid interfaces; effective simulations of multiple moving bodies and particulate matter within fluid flows; direct estimation of Lagrangian derivatives (absence of non-linear advective terms in the balance equations); effective numerical simulation of fast transient phenomena; no meshing; simple non-iterative algorithms (in case the "Weakly Compressible" approach is adopted). On the other hand, SPH models are affected by the following drawbacks, if compared with mesh-based CFD tools: computational costs are slightly higher due to a larger stencil (around each computational particle), which causes a high number of interacting elements (neighbouring particles) at a fixed time step (nonetheless SPH codes are more suitable to parallelization); local refining of spatial resolution represents a current issue and is only addressed by few advanced and complex SPH algorithms (e.g., Vacondio et al. 2016, [14]); accuracy is relatively low for classical CFD applications where mesh-based methods are well established (e.g., confined mono-phase flows). Detailed reviews on SPH assets and drawbacks have been reported in Gomez-Gesteira et al. (2010, [15]), Le Touzé et al. (2013, [16]), Shadloo et al. (2016, [17]),Violeau and Rogers (2016, [18]), Gotoh and Khayyer (2018, [19]), Manenti et al. (2019, [20]). Nevertheless, SPH models are effective in several, but peculiar, application fields. Some of them are herein briefly recalled: flood propagation (e.g., Gu et al. 2017, [21]; Vacondio et al. 2012, [22]; Crespo et al. 2008, [23]; Khayyer and Gotoh, 2010, [24]; Vacondio et al. 2014, [25]); sloshing tanks (e.g., Khayyer et al. 2018, [26]; Amicarelli et al. 2013, [5]); gravitational surface waves (e.g. Colagrossi et al. 2013, [27]; Crespo et al. 2007, [28]; Zhang et al. 2018, [29]) and marine energy converters (Crespo et al. 2017, [30]); hydraulic turbines (e.g., Marongiu et al. 2010, [31]); liquid jets (e.g., Marongiu et al. 2010, [31]); astrophysics and magneto-hydrodynamics (e.g., Price, 2012, [32]); body dynamics in free surface flows (e.g., Cercos-Pita et al. 2016, [33]; Cercos-Pita et al. 2017, [34]; Dominguez et al. 2019, [35]; Amicarelli et al. 2015, [4]); multi-phase and multi-fluid flows (Colagrossi & Landrini, 2003, [36]; Hu & Adams, 2006, [37]; Fonty et al. 2019, [38]); sediment removal from water reservoirs (e.g., Manenti et al. 2012, [6]); landslides (e.g., Abdelrazek et al. 2016, [39]; Bui et al. 2008, [40]). Some advanced multi-purpose SPH codes are available as FOSS (e.g., Crespo et al. 2015, [41]).

2. Scheme for transport of solid bodies and the semi-analytic approach as a boundary treatment scheme for fixed boundaries

This section describes the balance equations for fluid (Section 2.1) and body (Section 2.2) dynamics, the 2-way interaction terms related to both fluid–body (Section 2.3) and solid–solid (Section 2.4) interactions, the semi-analytic approach for treating fixed boundaries (Section 2.1).

2.1. SPH approximation of the balance equations for fluid dynamics and the boundary treatment scheme called "semi-analytic approach"

The numerical scheme for the main flow is a Weakly-Compressible (WC) SPH model, which takes benefit from a boundary treatment for fixed boundaries based on the semi-analytic approach of Vila (1999, [42]), as developed by Di Monaco et al. (2011, [7]).

One considers Euler's momentum and continuity equations:

$$\begin{aligned} \frac{du_i}{dt} &= -\frac{1}{\rho}\frac{\partial p}{\partial x_i} - \delta_{i3} g, \qquad i = 1, 2, 3 \\ \frac{d\rho}{dt} &= -\rho \nabla \cdot \underline{u} \end{aligned} \tag{2.1}$$

where $\underline{u} \equiv (u, v, w)$ (m/s) is the velocity vector, p (Pa) is the pressure, ρ (kg/m^3) is the fluid density, ρ_{ij} is Kronecker's delta function and t (s) denotes time. One needs to compute (2.1) at each fluid particle position by using the SPH formalism and by taking into account the boundary terms (fluid-frontier and fluid–body interactions), as described below.

One considers the discretization of (2.1), as provided by the SPH approximation of the first derivative of a generic function (f) – a variant of (1.3) – according to the semi-analytic approach ("$_{SA}$"; Vila, 1999, [42]):

$$\left\langle \frac{\partial f}{\partial x_i} \right\rangle_{SA,0} = -\sum_b (f_b - f_0) \frac{\partial W_b}{\partial x_i} \omega_b - \int_{V_h'} (f - f_0) \frac{\partial W}{\partial x_i} dx^3 \tag{2.2}$$

The inner fluid domain here involved is filled with numerical particles. At boundaries, the kernel support is (formally) not truncated because it can partially lie outside the fluid domain. In other words, the summation in (2.2) is performed over all the fluid particles "$_b$" (neighbouring particles with volume ω) in the kernel support of the computational fluid particle ("$_0$"). At the same time, the volume integral in (2.2) represents the boundary term, which is a convolution integral on the truncated portion of the kernel support $V_h{}'$ (m^3). In this fictitious and outer volume, one needs to define the generic function f (pressure, velocity or density alternatively).

The semi-analytic approach ("$_{SA}$") (as developed by Di Monaco et al. 2011, [7]) introduces the following linearization and assumptions to compute f in $V_h{}'$:

$$\begin{aligned} \left\langle \frac{\partial f}{\partial x_i} \right\rangle_{SA} = &-\sum_b (f_b - f_0) \frac{\partial W_b}{\partial x_i} \omega_b - \int_{V_h'} (f_{SA} - f_0) \frac{\partial W}{\partial x_i} dx^3 \\ &- \int_{V_h'} \left.\frac{\partial f}{\partial x_i}\right|_{SA} (\underline{x} - \underline{x}_0) \frac{\partial W}{\partial x_i} dx^3 \end{aligned} \tag{2.3}$$

The peculiar "$_{SA}$" values of the functions and derivatives within $V_h{}'$ are assigned to represent a null normal gradient of reduced

pressure at the frontier interface (while considering uniform density):

$$p_{SA} = p_0, \qquad \left\langle \frac{\partial p}{\partial x_i} \right\rangle_{SA} = -\delta_{i3} g; \qquad \rho_{SA} = \rho_0, \qquad \left\langle \frac{\partial \rho}{\partial x_i} \right\rangle_{SA} = 0 \tag{2.4}$$

At the same time, the model sets free-slip conditions when estimating velocity at boundaries. The velocity vector is taken as uniform in the outer part of the kernel support. Here $\underline{u}_{SA}$ is decomposed into the sum of a vector normal to boundary $\left(\underline{u}_{SA,n}\right)$ and a tangential vector $\left(\underline{u}_{SA,T}\right)$. The first is represented as a linear extrapolation from the computational fluid particle velocity. The latter is equal to its analogous vector of the same computational fluid particle (the subscript "$_w$" refers to a generic frontier), in case of free-slip conditions:

$$\left.\begin{aligned} &\underline{u}_{SA} = \underline{u}_{SA,T} + \underline{u}_{SA,n} = u_{0,T} + \left[\left(2\underline{u}_w - \underline{u}_0\right) \cdot \underline{n}\right] \underline{n} \\ &\underline{u}_{SA,T} = \underline{u}_{0,T}, \qquad \left\langle \frac{\partial u_i}{\partial x_i} \right\rangle_{SA} = 0 \end{aligned}\right\} \\ \Rightarrow \underline{u} - \underline{u}_0 = \underline{u}_{SA} - \underline{u}_0 = 2\left[\left(\underline{u}_w - \underline{u}_0\right) \cdot \underline{n}\right] \underline{n} \tag{2.5}$$

where $\underline{n}$ is the normal vector of the wall surface, as defined by its local orientation.

At this point, one can write the continuity equation for a Weakly Compressible SPH model (Einstein's notation is herein adopted for the subscript "$_j$"), using the semi-analytic approach for the boundary integral term (second term on the Right Hand Side):

$$\left\langle \frac{d\rho}{dt} \right\rangle_0 = \sum_b \rho_b \left(u_{b,j} - u_{0,j}\right) \left. \frac{\partial W}{\partial x_j} \right|_b \omega_b \\ + 2\rho_0 \int_{V_h'} \left[\left(\underline{u}_w - \underline{u}_0\right) \cdot \underline{n}\right] n_j \frac{\partial W}{\partial x_j} dx^3 + C_s \tag{2.6}$$

where C_s (kg $\times$ m$^{-3}\times$ s^{-1}) is introduced to represent a fluid–body interaction term.

On the other hand, one can analogously derive the approximation of the momentum equation (the notation $\langle\ \rangle$ indicates the SPH particle/discrete approximation):

$$\left\langle \frac{du_i}{dt} \right\rangle_0 = -\delta_{i3} g + \sum_b \left(\frac{p_b}{\rho_b^2} + \frac{p_0}{\rho_0^2} \right) \left. \frac{\partial W}{\partial x_i} \right|_b m_b + \\ + 2\frac{p_0}{\rho_0} \int_{V_h'} \frac{\partial W}{\partial x_i} dx^3 - \upsilon_M \sum_b \frac{m_b}{\rho_0 r_{0b}^2} \left(\underline{u}_b - \underline{u}_0\right) \cdot \left(\underline{x}_b - \underline{x}_0\right) \left. \frac{\partial W}{\partial x_i} \right|_b + \\ - 2\upsilon_M \left(\underline{u}_w - \underline{u}_0\right) \cdot \int_{V_h'} \frac{1}{r_{0_w}^2} \left(\underline{x} - \underline{x}_0\right) \left. \frac{\partial W}{\partial x_i} \right|_b dx^3 + \underline{a}_s \\ + 2\upsilon_0 \left(\underline{u}_w - \underline{u}_0\right) \int_{V_h'} \frac{1}{r} \left| \frac{\partial W}{\partial r} \right|_b dx^3 \tag{2.7}$$

where $\underline{a}_s$ (m $\times$ s^{-2}) represents a new acceleration term due to the fluid–body interactions, υ_M (m$^2\times$ s^{-1}) is the artificial viscosity (Monaghan, 2005, [8]), m (kg) is the particle mass and r (m) is the relative distance between the neighbouring and the computational particle.

Finally, a barotropic equation of state (EOS) is linearized as follows:

$$p \cong c_{ref}^2 \left(\rho - \rho_{ref}\right) \tag{2.8}$$

The artificial sound speed c (m/s) is 10 times higher than the maximum fluid velocity (WC approach) and "$_{ref}$" stands for a reference state.

More details are available in Amicarelli et al. (2015, [4]) and Di Monaco et al. (2011, [7]).

2.2. SPH balance equations for rigid body transport

Body dynamics is ruled by Euler–Newton equations, whose discretization takes advantage from the SPH formalism and the coupling terms derived in the following sections:

$$\frac{d\underline{u}_{CM}}{dt} = \frac{\underline{F}_{TOT}}{m_B}, \qquad \frac{d\underline{x}_{CM}}{dt} = \underline{u}_{CM} \\ \frac{d\underline{\chi}_B}{dt} = \underline{\underline{I}}_C^{-1} \left[\underline{M}_{TOT} - \underline{\chi}_B \times \left(\underline{\underline{I}}_C \underline{\chi}_B \right) \right], \qquad \frac{d\underline{\alpha}}{dt} = \underline{\chi}_B \tag{2.9}$$

Here the subscript "$_B$" refers to a generic computational body and "$_{CM}$" to its centre of mass. The first two formulas of (2.9) represent the balance equations for the momentum and the time law for the position of the body barycentre -$\underline{F}_{TOT}$ (kg $\times$ m $\times$ s^{-2}) being the global/resultant force exerted on the solid. The last two formulas of (2.9) express the balance equation of the angular momentum ($\underline{\chi}_B$ (rad $\times$ s^{-1}) denotes the angular velocity of the generic body) and the time evolution of the solid orientation ($\underline{\alpha}$ (rad) is the vector of Euler angles lying between the body axes and the global reference system). $\underline{M}_{TOT}$ (kg $\times$ m$^2\times$s^2) represents the associated torque acting on the body and $\underline{\underline{I}}_C$(kg$\times$m^2) the matrix of the moments of inertia of the computational body (Einstein's notation applied for the subscript "$_l$"):

$$I_{c,ij} = \int_{V_B}^{\rho} \left(r_l^2 \delta_{ij} - r_i r_j\right) dV = \begin{cases} \displaystyle\int_{V_B}^{\rho} \left(r_k^2 + r_n^2\right) dV, i = j;\ k, n \neq i \\ \displaystyle -\int_{V_B}^{\rho} \left(r_i r_j\right) dV, i \neq j \end{cases} \tag{2.10}$$

In this sub-section, $\underline{r}$ implicitly represents the relative distance from the body centre of mass.

In order to solve the system (2.9), we need to model the global force and torque, as described in the following.

The resultant force is composed of several terms:

$$\underline{F}_{TOT} = \underline{G} + \underline{P}_F + \underline{T}_F + \underline{P}_S + \underline{T}_S, \qquad \underline{T}_F + \underline{T}_S \cong 0 \tag{2.11}$$

where $\underline{G}$ (kg $\times$ m $\times$ s^{-2}) represents the gravity force, whereas $\underline{P}_F$ (kg $\times$ m $\times$ s^{-2}) and $\underline{T}_F$ (kg $\times$ m $\times$ s^{-2}) the vector sums of the pressure and shear forces provided by the fluid. Analogously, $\underline{P}_S$ (kg $\times$ m$\times$ s^{-2}) and $\underline{T}_S$ (kg $\times$ m $\times$s^{-2}) are the vector sums of the normal and the shear forces provided by other bodies or boundaries (solid–solid interactions). In case of inertial and quasi-inertial fluid flows, we do not need to refer to either turbulence scheme or tangential stresses (simplifying hypothesis).

The fluid–solid interaction is expressed by the hydrodynamic thrust:

$$\underline{P}_F = \sum_s p_s A_s \underline{n}_s \tag{2.12}$$

The computational body is numerically represented by solid volume elements, here called (solid) "body particles" ("$_s$"). Some of them describe the body surface and are referred to as "surface body particles". These particular elements are also characterized by an area and a vector $\underline{n}$ of norm 1. This is perpendicular to the body face of the particle (it belongs to) and points outward the fluid domain (inward the solid body). Graphical and in-depth details are available in Amicarelli et al. (2015, [4]).

The pressure of a body particle is computed as described in Section 2.3, whereas the treatment of the solid–solid interaction term ($\underline{P}_s$) is discussed in Section 2.4.

The torque in (2.9) is discretized as the summation of each vector product between the relative position $\underline{r}_s$, of a surface

body particle with respect to the body centre of mass, and the corresponding total particle force:

$$\underline{M}_{TOT} = \sum_s \underline{r}_s^{\times} \underline{F}_s \tag{2.13}$$

Time integration of (2.9) is performed using a Leapfrog scheme synchronized with the fluid dynamics balance equations. This means that the body particle pressure is computed simultaneously to the fluid pressure, so that this parameter is staggered of around $dt/2$ (s) with respect to all the other body particle parameters.

After time integration, the model obtains the velocity of a body particle as the vector sum of the velocity of the corresponding body barycentre and the relative velocity:

$$\underline{u}_s = \underline{u}_{CM} + \underline{\chi}_B \times \underline{r}_s \tag{2.14}$$

Finally, the model updates the body particle normal vectors and absolute positions, according to the following kinematics formulas -$\underline{d\alpha}$ (rad) is the vector increment in Euler's angles during the current time step and R_{ij} is the body rotation matrix-:

$$\begin{aligned}
&\underline{n}_s(t+dt) = \underline{\underline{R}}_B \underline{n}_s(t), \qquad \underline{x}_s(t+dt) = \underline{x}_{CM}(t+dt) + \underline{\underline{R}}_B \underline{r}_s(t) \\
&\underline{\underline{R}}_B = \underline{\underline{R}}_x \underline{\underline{R}}_y \underline{\underline{R}}_z, \qquad \underline{d\alpha}_B = \underline{\chi}_B dt \\
&\underline{\underline{R}}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(d\alpha_x) & -\sin(d\alpha_x) \\ 0 & \sin(d\alpha_x) & \cos(d\alpha_x) \end{bmatrix}, \\
&\underline{\underline{R}}_y = \begin{bmatrix} \cos(d\alpha_y) & 0 & \sin(d\alpha_y) \\ 0 & 1 & 0 \\ -\sin(d\alpha_y) & 0 & \cos(d\alpha_y) \end{bmatrix}, \\
&\underline{\underline{R}}_z = \begin{bmatrix} \cos(d\alpha_z) & -\sin(d\alpha_z) & 0 \\ \sin(d\alpha_z) & \cos(d\alpha_z) & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{2.15}$$

More details are available in Amicarelli et al. (2015, [4]).

2.3. Fluid–body interaction terms

The fluid–body interaction terms rely on the boundary technique introduced by Adami et al. (2012, [9]), as implemented and adapted for free-slip conditions by Amicarelli et al. (2015, [4]). If boundary is fixed, this method can be interpreted as a discretization of the semi-analytic approach used to treat fluid–boundary interactions (Section 2.1). The outer domain of (2.2) is herein represented by all the body particles inside the kernel support of the computational fluid particle. Further, Adami et al. (2012, [9]) introduced a new term, related to the acceleration of the fluid–solid interface, which influences the estimation of body particle pressure.

The fluid–body interaction term in the continuity equation represents a discrete approximation of the analogous term in (2.6), used to treat solid frontiers (free-slip conditions):

$$C_s = 2\rho_0 \sum_s \left[(\underline{u}_s - \underline{u}_0) \cdot \underline{n}_s \right] W_s' \omega_s \tag{2.16}$$

Analogously, the fluid–body interaction term in the momentum equation (2.7) assumes the form:

$$\underline{a}_s = \sum_s \left(\frac{p_s + p_0}{\rho_0^2} \right) W_s' m_s \tag{2.17}$$

The pressure value of the generic neighbouring surface body particle "$_s$" depends on the particular computational fluid particle "$_0$" we are considering, so that we can refer to the interaction subscript "$_{s,0}$". One may apply a SPH interpolation over all the pressure values coming from fluid–body particle interactions to derive a unique pressure value for each body particle (free-slip conditions):

$$p_s = \frac{\sum_0 \left\{ p_0 + \rho_0 \left[(\underline{g} - \underline{a}_s) \cdot \underline{n}_w \right] \left[(\underline{x}_s - \underline{x}_0) \cdot \underline{n}_w \right] \right\} W_{0s} \left(\frac{m_0}{\rho_0} \right)}{\sum_0 W_{0s} \left(\frac{m_0}{\rho_0} \right)} \tag{2.18}$$

More details are available in Amicarelli et al. (2015, [4]).

2.4. Solid–solid interaction terms

The solid–solid interaction term $\underline{P}_s$ in (2.11) represents body–body and body–boundary impingement forces, whose time and spatial evolution, in the continuum, are theoretically proportional to Dirac's delta function. The numerical model needs to discretize $\underline{P}_s$, as explained hereafter.

The "boundary force particle" method of Monaghan (2005, [8]) defines repulsive forces to represent a conservative full elastic impingement between two SPH interacting particles (of any medium), conserving both global momentum and kinetic energy. The formulation above applies for inter-particle high velocity impacts and is implemented and extended to whole solid bodies (not only particle impingements), even at low velocities, as well as body-frontier interactions (Amicarelli et al. 2015, [4]).

One considers the overall force $\underline{P}_s$, which represents the impingements between a generic computational body ("$_B$") and all its neighbouring bodies ("$_K$") and frontiers ("$_{K^*}$"). $\underline{P}_s$ is decomposed in elementary 2-body ($\underline{P}_{BK}$) and body-frontier ($\underline{P}_{BK^*}$) interactions:

$$\underline{P}_S = \sum_K \underline{P}_{BK} + \sum_{K^*} \underline{P}_{BK^*} \tag{2.19}$$

Invoking the same principles of the boundary force particle method, $\underline{P}_{BK}$ involves interactions between all the body particles "$_j$" of the computational body "$_B$" and their neighbour body particles "$_k$", belonging to the neighbouring body "$_K$":

$$\underline{P}_{BK} = -\alpha_I \sum_j \sum_k \frac{2u_{\perp,jk}^2}{r_{per,jk}} \frac{m_j m_k}{m_j + m_k} \Gamma_{jk} \left(1 - \frac{r_{par,jk}}{dx_s} \right) \underline{n}_k \tag{2.20}$$

The components of the inter-particle relative distance, $\underline{r}_{par}$ and $\underline{r}_{per}$, are parallel and perpendicular to the neighbour normal, respectively. The term within brackets in (2.20) deforms the kernel support of the body particles "$_j$", so that it mainly develops along the direction aligned with the normal of the neighbouring particle (dx_s (m) is the size of the solid body particles). The weighting function Γ is expressed according to Monaghan (2005, [8]) and depends on $q = r_{jk}/h$:

$$\Gamma_{jk} = \begin{cases} \frac{2}{3}, & 0 \le q < \frac{2}{3} \\ \left(2q - \frac{3}{2} q^2 \right), & \frac{2}{3} \le q < 1 \\ \frac{1}{2}(2-q)^2, & 1 \le q < 2 \\ 0, & 2 \le q \end{cases} \tag{2.21}$$

SPHERA holds two modifications for body–body interactions, with respect to the original formulation of the boundary force particles. The first one concerns the impact velocity $u_{\perp,jk}$ (m/s), which replaces the term "0.1c" in the formulation of Monaghan (2005, [8]) and properly deals with low velocity impacts. It avoids too strong or too weak impingement forces. For each body–body

interaction, the impact velocity has a unique value for all the particle–particle interactions during the on-going time step. This velocity is computed as the maximum of the absolute values of the inter-particle relative velocity (projected over the normal of the neighbouring particle). For this purpose, the scheme considers all the inter-particle interactions recorded while the 2 bodies are approaching. The expression for the impact velocity reads:

$$u_{\perp,jk}(t) = \max_{j,k,t^*} \left\{ \left| \left[\left(\underline{u}_j - \underline{u}_k \right) \cdot \underline{n}_k \right] \right| \right\}, \qquad t_0 \leq t^* \leq t \tag{2.22}$$

where t_0 (s) refers to the beginning of the approaching phase. When other forces (e.g. pressure and gravity forces) are taken into account, the impact velocity can eventually increase in the inter-body impact zone, causing a potential and partial penetration of a solid into another body. Under these conditions, and only during the approaching phase, (2.22) allows increasing the magnitude of the impingement force, depending on the actual impact velocity (instead of the undisturbed impact velocity). This modification avoids mass penetrations in case of complex impingements.

Further, (2.20) introduces the normalizing parameter α_I, which corrects discretization errors and better preserves the global momentum and kinetic energy of the body–body system during the impingement. If one omits α_I, then (2.20) would drastically under-estimate the impingement forces if the whole mass of the bodies does not lie within the impact zone (of depth $2h$). The expression for α_I assumes the following form:

$$\alpha_I = \frac{1}{r_{per,BK}} \frac{m_B m_K}{m_B + m_K} \Gamma_{BK} \Bigg/ \sum_j \sum_k \left[\frac{1}{r_{per,jk}} \frac{m_j m_k}{m_j + m_k} \Gamma_{jk} \left(1 - \frac{r_{par,jk}}{dx_s} \right) \right] \tag{2.23}$$

where the body impact velocity is represented as a weighted average of the particle impact velocities. As a first approximation, the normalizing factor α_I roughly represents the inverse of the fraction of the system mass which lies in the impingement zone. This mass should numerically represent the whole 2-body system during the impact.

Finally, the model represents body–boundary interactions. A generic boundary is modelled as a body with infinite mass and discretization tending to zero (the semi-analytic approach, used to model frontiers, is an integral method). The interaction force assumes the following expression (here the subscript "$_{K^*}$" refers to a generic neighbouring frontier):

$$\underline{P}_{BK^*} = -\alpha_I \sum_j \frac{2u^2_{\perp,jK^*}}{r_{per,jK^*}} m_j \Gamma_{jK^*} \underline{n}_{K^*},$$
$$\alpha_I = \frac{m_B}{r_{per,BK^*}} \Gamma_{BK^*} \Bigg/ \sum_j \left(\frac{m_j}{r_{per,jK^*}} \Gamma_{jK^*} \right) \tag{2.24}$$

More details are available in Amicarelli et al. (2015, [4]).

3. Scheme for dense granular flows

This section describes the mathematical and numerical models of the scheme for dense granular flows (Amicarelli et al. 2017, [3]; Section 3.1). This mixture model for bed-load transport and fast landslides is consistent with the "packing limit" of the Kinetic Theory of Granular Flow (KTGF, Armstrong et al. 2010, [43]) and no tuning parameter is used to represent the mixture viscosity. In case erosion is the only cause of mobilization of the solid grains, the model above can be possibly sped-up in 2D by means of an erosion scheme (Manenti et al. 2012, [6]; Section 3.2).

3.1. Mixture model for dense granular flows

This SPH scheme represents the mixture of fluid phase and non-cohesive solid granular material, under the "packing limit" of the Kinetic Theory of Granular Flow (KTGF, Armstrong et al. 2010, [43]) for dense granular flows. This limit refers to the maximum values of the solid phase volume fraction and is peculiar of bed-load transport (e.g., erosional dam breaks) and fast landslides.

Adopting a Weakly Compressible approach, the continuity equation for the mixture reads:

$$\frac{d\rho}{dt} = -\rho \frac{\partial u_j}{\partial x_j} \tag{3.1}$$

The mixture density and velocity (the subscript "$_m$" for the mixture quantities are omitted for simplicity of notation) are defined as follows:

$$\rho \equiv \rho_f \varepsilon_f + \rho_s \varepsilon_s, \qquad u_i \equiv \frac{\rho_f \varepsilon_f u_{f,i} + \rho_s \varepsilon_s u_{s,i}}{\rho}, \qquad i = 1, 3 \tag{3.2}$$

The volume fractions (ε) of the fluid ("$_f$") and the solid ("$_s$") phases are constrained to the volume balance equation:

$$\varepsilon_s + \varepsilon_f = 1 \tag{3.3}$$

The model assumes that SPH particles are conservative (i.e. mixture particles do not exchange net mass fluxes with the surrounding environment), which is a reasonable hypothesis for high solid volume fractions in saturated soils, according to the "packing limit" of the Kinetic Theory of Granular Flow (Armstrong et al. 2010, [43]).

Following the multi-phase approach of Colagrossi and Landrini (2003, [36]), the SPH approximation of (3.1) can be expressed as follows:

$$\frac{d\rho_0}{dt} = \rho_0 \sum_b \left(u_{b,j} - u_{0,j} \right) \left. \frac{\partial W}{\partial x_j} \right|_b \omega_b + 2\rho_0 \int_{V'_h} \left[\left(\underline{u}_w - \underline{u}_0 \right) \cdot \underline{n} \right] n_j \frac{\partial W}{\partial x_j} dx^3 \tag{3.4}$$

The integral boundary term in (3.4) is computed according to Di Monaco et al. (2011, [7]) and represents the effects of wall frontiers.

The form of the momentum equations for the mixture is identical to Navier–Stokes equations for incompressible fluids:

$$\frac{du_i}{dt} = -\delta_{i3} g - \frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j^2} \tag{3.5}$$

where p (Pa) is the mixture total pressure/stress and ν ($\mathrm{m^2 \times s^{-1}}$) is the mixture kinematic viscosity.

The total stress is computed by means of a barotropic equation of state, formally identical to (2.8). A unique speed of sound can be chosen (i.e. the highest among the SPH particle values, no matter about their phase volume fractions).

The mixture dynamic viscosity ($\mu \equiv \upsilon \rho$) (Pa $\times$ s) is defined as:

$$\mu \equiv \varepsilon_f \mu_f + H\left(\varepsilon_s - \varepsilon_{s,p} \right) \mu_{fr} \tag{3.6}$$

where H is the Heaviside step function.

In the "packing limit" of the KTGF (i.e. for ε_s close enough to the value of $\varepsilon_{s,p}$ = ca.0.59, which is the maximum attainable solid volume fraction for a sheared inelastic hard sphere fluid, Kumaran, 2015, [44]), the shear stress gradient term is represented by means of a plastic model for dry granular material based on internal friction (Schaeffer, 1987, [45]), by means of a physical quantity named frictional viscosity μ_{fr} (Pa $\times$ s):

$$\mu_{fr} \equiv \left(\frac{\sigma'_m (\sin \varphi)}{2\sqrt{I_2 (e_{ij})}} \right) \tag{3.7}$$

Here ϕ (rad) is the internal friction angle, e_{ij} (s^1) the strain-rate tensor and $I_2(e_{ij})$ (s^{-2}) is its second invariant (formulation for incompressible fluids):

$$e_{ij} \equiv \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \quad |e_{ij}| \equiv \left(\sum_{i,j} e_{ij}^2\right)^{\frac{1}{2}} = \sqrt{2I_2\left(e_{ij}\right)} \tag{3.8}$$

The mean effective stress σ'_m(Pa) is computed as the difference between the total stress and the fluid pressure, according to the principle of Terzaghi (1943, [46]):

$$p = p_f + \sigma'_m \tag{3.9}$$

The fluid pressure in the granular material is related to two different soil conditions, as follows:

$$p_f = \begin{cases} p_{f,blt-top} + \rho_f g(z_{blt-top}|_{x_0,y_0} - z_0)\cos^2(\alpha_{TBT}), \\ \quad fully \quad saturated \quad soil \\ 0, \quad dry \quad soil \end{cases} \tag{3.10}$$

where the subscript "$_{blt-top}$" refers to the top of the bed-load transport layer (or the layer of saturated material). Eq. (3.10) assumes a 1D filtration flow parallel to the slope of the granular material. This simplifying hypothesis is still consistent with SPH conservative particles; α_{TBT} (rad) is the topographic angle at the top of the bed-load transport layer and lies between the local interface normal and the vertical.

Following the multi-phase approach of Colagrossi and Landrini (2003, [36]), with the boundary treatment method proposed by Di Monaco et al. (2011, [7]), the SPH approximation of the momentum equations (3.5) becomes:

$$\begin{aligned} \left\langle \frac{du_i}{dt} \right\rangle_0 &= -\delta_{i3}g + \frac{1}{\rho_0}\sum_b (p_b + p_0)\left.\frac{\partial W}{\partial x_i}\right|_b \omega_b \\ &+2\frac{p_0}{\rho_0}\int_{V'_h}\left.\frac{\partial W}{\partial x_i}\right|_b dx^3 + 2\upsilon\sum_b \frac{m_b}{\rho_0 r_{0b}}\left(\underline{u}_b - \underline{u}_0\right)\left.\frac{\partial W}{\partial r}\right|_b + \\ &-\upsilon_M \sum_b \frac{m_b}{\rho_0 r_{0b}^2}\left(\underline{u}_b - \underline{u}_0\right)\cdot\left(\underline{x}_b - \underline{x}_0\right)\left.\frac{\partial W}{\partial x_i}\right|_b \\ &-\upsilon_M\left(\underline{u}_{SA} - \underline{u}_0\right)\cdot\left(\int_{V'_h}\frac{1}{r_{0w}^2}\left(\underline{x} - \underline{x}_0\right)\frac{\partial W}{\partial x_i}dx^3\right) \end{aligned} \tag{3.11}$$

In order to pinpoint the elasto-plastic regime and avoid the unbounded growth of (3.7), the threshold ν_{max} ($m^2 \times s^{-1}$) for the mixture viscosity is introduced. Mixture particles with a higher viscosity value are considered in the elasto-plastic regime of soil deformation. As their displacements are negligible, these particles are kept fixed as long as they belong to this regime and their pressure is derived from the mixture particles flowing above them. The viscosity threshold is assumed to be high enough not to influence the simulation results.

Further details are available in Amicarelli et al. (2017, [3]).

3.2. 2D erosion criterion

An erosion criterion (Manenti et al. 2012, [6]) is implemented to speed-up the 2D simulations of the scheme for dense granular flows (Section 3.1), only for those configurations where erosion is the only cause of mobilization of the solid grains. The erosion criterion aims to select those mixture particles, which needs the scheme for dense granular flows (Section 3.1) to be applied.

The 2D erosion scheme considers the interface "pure fluid – fixed bed" and it is based on the formulation of Shields – van Rijn (1993, [47]). The erosion criterion refers to the interaction of a generic fixed mixture particle and the pure fluid above. Its reference quantities are represented by the closest mobile particle (of pure fluid) above the fixed mixture particle.

The erosion criterion is satisfied if the Shields parameter (θ), which is defined as:

$$\vartheta \equiv \frac{\tau_*}{g\left(\rho_s - \rho_f\right)d_{50}}, \qquad \tau_* \equiv \rho_f u_*^2 \tag{3.12}$$

is equal or greater than its critical value θ_c, expressed by van Rijn (1993, [47]):

$$\vartheta_c = \begin{cases} 0.010595\ln(\mathrm{Re}_*) + \frac{0.110476}{\mathrm{Re}_*} + 0.0027197, & \mathrm{Re}_* \leq 500 \\ 0.068, & \mathrm{Re}_* > 500 \end{cases} \tag{3.13}$$

Re_* is defined as the grain Reynolds number:

$$\mathrm{Re}_* \equiv \frac{d_{50}u_*}{\upsilon_f} \tag{3.14}$$

where d_{50} (m) is the 50th percentile of the particle-size distribution of the soil.

If the height of the liquid particle $z = x_3$ belongs to the Surface Neutral Boundary Layer (SNBL), then the friction velocity u_* (m/s) is iteratively computed according to the similarity theory for the SNBL and the formula for the roughness coefficient z_0 (m) generalized by Manenti et al. (2012, [6]):

$$u_* = \frac{k_\upsilon U}{\ln\left(z/z_0\right)}, \qquad z_0 = 0.11\frac{\upsilon_f}{u_*} + \frac{d_{50}}{30} \tag{3.15}$$

where k_υ is von Kármán constant and U (m/s) the flow velocity at the height of the liquid particle.

If the liquid particle lies below the SNBL, then the model considers the velocity profile of the Sub-Viscous Layer, with the following direct estimation for the friction velocity:

$$u_* = \sqrt{\frac{U\upsilon_f}{z}} \tag{3.16}$$

The 2D erosion criterion is derived under 1D stationary and uniform conditions, and does not explicitly depends on the internal friction angle of the granular material.

Graphical and further in-depth details are available in Manenti et al. (2012, [6]).

4. DB-SPH boundary treatment scheme

The activation of the "Discrete Boundary" (DB)-SPH method for boundary treatment (Amicarelli et al. 2013, [5]) allows to treat fixed and mobile solid boundaries and also alters the balance equations in the internal domain of Section 2.1. The following sub-sections briefly describe the DB-SPH particle approximation (Section 4.1), the modifications of the balance equations in the inner domain (Section 4.1) and the 1D Linearized Partial Riemann Solver associated with DB-SPH (Section 4.2).

4.1. DB-SPH particle approximation and modifications of the balance equations in the inner domain

According to the DB-SPH method, the first derivative of a generic function (f) is approximated by means of the following SPH particle approximation:

$$\begin{aligned} \left\langle \left.\frac{\partial f}{\partial x_i}\right| \right\rangle_{\underline{x}_0} &\equiv \sum_a (f_A)\frac{W_A}{\gamma_0}n_{i,a}\omega_a - \sum_b (f_b)\frac{W'_b}{\gamma_0}\omega, \\ W'_b &\equiv \left.\frac{\partial W}{\partial x_i}\right|_b, \qquad \gamma \equiv \int_{V_h} W dx^3 \end{aligned} \tag{4.1}$$

The volume integral of (1.2) is here replaced with a summation over the fluid particles within the kernel support. The surface integral of (1.2) is herein replaced with a summation over the wall

surface elements "$_a$" intercepted by the kernel support volume V_h, which is normalized by the integral Shepard coefficient γ. This normalization allows considering the truncated kernel support as if it were entire (in the continuum), but with non-spherical shape. Eq. (4.1) is used to approximate the pressure gradient term of Euler momentum equations (2.1).

The DB-SPH scheme adopts semi-particles, whose 3D definition is slightly different from the edge particles (semi-particles) of Ferrand et al. (2013, [13]). The "semi-particles" represent special fluid particles, which are smallest than the (inner) fluid particles. Each semi-particle is associated to a surface wall element. Semi-particle positions are formally located at the solid frontiers of the fluid domain, but the volumes of the semi-particles completely lie in the inner domain and touch the solid boundaries. The union of the semi-particle volumes represents a thin film of fluid, which is a buffer zone between the inner domain (filled with computational particles) and the wall frontiers. The film depth is smaller than the characteristic length of the fluid particles dx (m).

Surface elements and semi-particles share the same values of their physical quantities. Every surface element is defined by its position, velocity vector, area (length in 2D) and normal vector. Semi-particles additionally require the mass.

When activating the DB-SPH boundary treatment, a SPH particle approximation of density replaces the continuity equation in the inner domain (Ferrand et al. 2013, [13]):

$$\langle \rho \rangle_0 = \frac{\sum_{bs} \rho_{bs} W_{bs} \omega_{bs}}{\tilde{\gamma}_0} \tag{4.2}$$

where the kernel is normalized by a corrected estimation of the integral Shepard coefficient and the subscript "$_s$" refers to the semi-particles.

The following correction of γ avoids excessive SPH truncation errors at the free surface:

$$\tilde{\gamma} \equiv \begin{Bmatrix} \sigma, & \gamma \geq \sigma + \sigma_\varepsilon \\ \gamma, & \gamma < \sigma + \sigma_\varepsilon \end{Bmatrix}, \qquad \sigma \equiv \sum_{bs} W_{bs} \omega_{bs} \tag{4.3}$$

The integral Shepard coefficient is replaced with the discrete Shepard coefficient at the free surface, which is numerically defined where $\gamma \geq \sigma + \sigma_\varepsilon$. The constant σ_ε can be set equal to 0.05 or chosen as an input parameter to better detect the free surface, depending on the test case and the spatial resolution.

A direct estimation of γ would imply the expensive estimation of 3D analytical integrals. Instead, the present model follows the procedure of Ferrand et al. (2013, [13]), who consider the Lagrangian derivative of γ:

$$\frac{d\gamma}{dt} \cong \sum_a W_a \underline{n}_a \cdot \left(\underline{u}_a - \underline{u}_0 \right) \omega_a, \qquad \frac{\partial W}{\partial t} = 0 \tag{4.4}$$

The initial values of γ are approximately provided by the associated values of σ, as the model exactly assigned the initial values of the fluid particle volumes:

$$\gamma_0 \left(t = 0 \right) \cong \begin{Bmatrix} 1, & \min \{ r_{0a} \} \geq 2h \\ \sigma_0 \left(t = 0 \right), & \min \{ r_{0a} \} < 2h \end{Bmatrix} \tag{4.5}$$

Further details are available in Amicarelli et al. (2013, [5]).

4.2. 1D Linearized Partial Riemann Solver

At boundaries, the fluid velocity component, which is perpendicular to the wall frontier, is equal to the same component of the frontier velocity (non-penetration condition). The model adopts a 1D LPRS (Linearized Partial Riemann Solver) to impose boundary conditions at the wall elements and semi-particles. The 1D LPRS is an up-wind scheme, also used in SPH-ALE modelling (Marongiu et al. 2010, [31]), which allows wall pressure being approximately compatible with the 3D pressure and velocity fields in the inner domain (constrained to the frontier kinematics).

The definition of the initial conditions ("^{L}", "Left") of the 1D LPRS is described by means of a first order MUSCL spatial reconstruction scheme. For each interaction ("$_{0a}$") between a surface element ("$_a$") and a fluid particle ("$_0$"), the LPRS initial conditions are defined at the position of the wall element. Here the model estimates density and the velocity components, by means of the MUSCL scheme around he computational particle (f alternatively refers to density and every velocity component):

$$f^L_{0a} \cong f_0 + \left\langle \underline{\nabla f} \right\rangle_0 \cdot \left\langle \underline{x}_a - \underline{x}_0 \right\rangle, \qquad u_{n,0a} = \underline{u}^L_{0a} \cdot \underline{n}_a \tag{4.6}$$

The velocity vector is projected along the normal of the surface wall element to obtain $\underline{u}_n$.

The solution (*) of the LPRS (at the wall element position) provides a reconstructed density value, whereas the associated pressure comes from the equation of state:

$$\rho^*_{0a} = \rho^L_{0a} + \left(u^L_{n,0a} - u_{n,a} \right) \frac{\rho^L_{0a}}{c^L_{0a}}, \qquad p_{0a} = c^2 \left(\rho^*_{0a} - \rho_0 \right) \tag{4.7}$$

So far, the scheme has estimated several pressure values, at each wall element. The following SPH approximation of these values (summation over all the neighbouring fluid particles) provides a unique pressure value for the surface element:

$$p_a = \frac{\sum_a p_{0a} \left(W \omega \right)_a}{\sum_a \left(W \omega \right)_a} \tag{4.8}$$

Further details are available in Amicarelli et al. (2013, [5]).

5. Time integration schemes

Time integration is ruled by a second-order Leapfrog scheme (stability analysis and time integration schemes in SPH modelling are discussed in Violeau and Leroy, 2014, [48]), as described in Di Monaco et al. (2011, [7]) and Amicarelli et al. (2015, [4]):

$$\begin{aligned} x_i|_{t+dt} &= x_i|_t + u_i|_{t+dt/2}\, dt, \qquad i = 1, 2, 3 \\ u_i|_{t+dt/2} &= u_i|_{t-dt/2} + \left\langle \frac{du_i}{dt} \right\rangle \Big|_t dt, \qquad i = 1, 2, 3 \\ \rho|_{t+dt} &= \rho|_t + \left\langle \frac{d\rho}{dt} \right\rangle \Big|_{t+dt/2} dt \end{aligned} \tag{5.1}$$

An alternative first-order Runge–Kutta time integration scheme is also available (Euler scheme).

Time integration is constrained by the following stability criteria:

$$dt = \min_0 \left\{ C_\nu \frac{2h^2}{\nu}; CFL \frac{2h}{c + |\underline{u}|} \right\} \tag{5.2}$$

where CFL is the Courant–Friedrichs–Lewy number. Following Adami et al. (2012, [9]), the viscous term stability parameter is set to $C_\nu = 0.05$.

6. Source code, executable code, input files

A synthetic description of the program units is reported in Section 6.1; the code compilation and execution are discussed in Section 6.2; the main input file of SPHERA is presented in Section 6.3. Further details are available on SPHERA (2018, [1]).

Table 6.1
Folders of SPHERA v.9.0.0 repository.

Folder	Description
(repository folder)	GNU-GPL license file
doc	Documentation file
src	Source code (with makefile)
bin	Executable files compiled with gfortran/ifort for optimized executions
debug	Executable files compiled with gfortran/ifort for debug scalar executions
debug_omp	Executable files compiled with gfortran/ifort for debug parallel executions
input	Input files for tutorials and a commented template for the main input file

Table 6.2
Program units (".f90") for the boundary conditions of the inlet/outlet sections (SPHERA, 2018, [1], folder "BC").

Program unit	Synthetic description
CancelOutgoneParticles_2/3D	To count and delete the outgoing particles at boundaries
FindFrame/Line	Extreme coordinates of the parallelepiped/rectangle containing the domain
GenerateSourceParticles_2/3D	To generate new source particles at the inlet sections
NormFix, NumberSectionPoints	Minor program units
PreSourceParticles_2/3D	To generate new source particles at the inlet sections (initial time)
VelLaw	To impose an input kinematics to particles

6.1. Synthetic description of the program units

The folders of SPHERA repository are reported in Table 6.1. The program units of SPHERA v.9.0.0 (folder "src") are grouped in subfolders, associated with the following topics: boundary conditions ("BC", Table 6.2); continuity equation (Table 6.3); momentum equation (Table 6.4); transport of solid bodies (Table 6.5); constitutive equation (Table 6.6); boundary treatment scheme "DB-SPH" (Table 6.8); erosion criterion (Table 6.7); geometry/algebra (Table 6.9); initial conditions ("IC", Table 6.10); main program ("main"), program units for the main code algorithms (both in 2D and 3D), memory management and Leapfrog time integration scheme (Table 6.11); Fortran modules (Table 6.12); neighbouring search, smoothing operators and interface detection (Table 6.13); post-processing (Table 6.14); pre-processing (Table 6.15); boundary treatment scheme "SA-SPH" (Table 6.16); managing Fortran character variables; Runge–Kutta time integration schemes (Table 6.17).

The input files of SPHERA are listed hereafter:

- main input file (user-defined name; the ".inp" format is defined in SPHERA, 2018, [1]);
- file list for the DB-SPH surface meshes ("surface_mesh_list.inp");
- ensemble of the files of the DB-SPH surface meshes (".ply" file format).

The main output files of SPHERA report the following information:

- application log of SPHERA;
- 3D fields of the main fluid dynamics and SPH variables (".vtu" and ".pvd" file formats) for Paraview visualization;
- frontier geometry for the boundary treatment SA-SPH (".vtk" format for Paraview);

Table 6.3
Program units (".f90") for the continuity equation (SPHERA, 2018, [1], folder "BE_mass").

Program unit	Synthetic description
CalcPre	To estimate particle pressure
Continuity_Equation	To assess continuity equation RHS, velocity gradients and strain-rate tensor.
inter_SmoothPres	To calculate a corrective term for pressure
PressureSmoothing_2/3D	To execute partial smoothing for pressure

Table 6.4
Program units (".f90") for the momentum equation (SPHERA, 2018, [1], folder "BE_momentum").

Program unit	Synthetic description
Diffumorris	Minor subroutine
inter_EqMoto	To assess the momentum equation RHS and DB-SPH terms
velocity_smoothing/_SA_SPH_2/3D	To assess a corrective term for velocity
viscomon	To assess an artificial viscosity term
viscomorris	To assess the viscous shear stress term in the momentum equation

Table 6.5
Program units (".f90") for the transport of solid bodies (SPHERA, 2018, [1], folder "Body_Transport").

Program unit	Synthetic description
Body_dynamics_output	To write ".txt" output files for body transport in fluid flows
body_particles_to_continuity	Contributions of the body particles to the continuity equation
body_pressure_mirror	Computation of the body particle pressure
body_pressure_postpro	Post-processing for body particle pressure
body_to_smoothing_pres	Contributions of body particles to pressure partial smoothing
body_to_smoothing_vel	Contributions of body particles to the possible velocity correction term
Gamma_boun	Interpolative function for boundary force particles
Input_Body_Dynamics	Advanced input management for body transport
RHS_body_dynamics	To estimate the RHS of the body dynamics equations

Table 6.6
Program units (".f90") for the constitutive equation (SPHERA, 2018, [1], folder "Constitutive_Equation").

Program unit	Synthetic description
Mixture_viscosity	To assess frictional viscosity and mixture viscosity for dense granular flows
Viscapp	Minor subroutine

Table 6.7
Program units (".f90") for the erosion criterion (SPHERA, 2018, [1], folder "Erosion_Criterion").

Program unit	Synthetic description
Compute_k_BetaGamma	Coefficient for Shields parameter in 3D
fixed_bed_slope_limited	Management of mixture particles belonging to the fixed bed
Shields	2-interface 3D erosion criterion

Table 6.8
Program units (".f90") for the boundary treatment scheme (SPHERA, [1], 2018, folder "DB_SPH").

Program unit	Synthetic description
adjacent_faces_isolated_points	Provided 2 adjacent faces, it finds the vertices not in common.
BC_wall_elements	To assess wall element density and pressure
DBSPH_BC_shear_viscosity_term	Contributions to the numerator of the shear viscosity term
DBSPH_find_close_faces	Finding the adjacent surface elements of a given surface element
DBSPH_IC_surface_elements	Initialization of wall surface elements
DBSPH_inlet_outlet	Imposing boundary conditions at the inlet and outlet sections
DBSPH_kinematics	Input kinematics for the DB-SPH elements (interpolation of input data)
DBSPH_velocity_gradients_VSL_SNBL	Velocity gradients in the Viscous Sub-Layer
Gradients_to_MUSCL	Consistency estimation of velocity and density gradients for MUSCL
Gradients_to_MUSCL_boundary	Boundary terms for the MUSCL reconstruction scheme
Import_ply_surface_meshes	Managing the surface meshes of SnappyHexMesh
semi_particle_volumes	Semi-particle shape coefficients and volumes
wall_elements_pp	Smoothing wall element physical quantities for post-processing
wavy_inlet	To provide a very slightly wavy flow at the inlet section

Table 6.9
Program units (".f90") on geometry/algebra (SPHERA, 2018, [1], folder "Geometry").

Program unit	Synthetic description
area_hexagon/…/triangle	Area of a generic hexagon/…/triangle from of its vertex coordinates
dis_point_plane	To assess the distance between a point and a plane
distance_point_line_2/3D	To assess the distance between a point and a line in 2/3D
IsPointInternal	To check whether a point is internal/external to a given face
line_plane_intersection	Intersection point (if unique) between a line and a plane
LocalNormalCoordinates	To compute the local normal vectors
Matrix_Inversion_2 × 2/3 × 3	To assess the inverse of a provided 2 × 2/3 × 3 matrix
MatrixProduct	To assess the matrix product between two matrices
MatrixTransposition	To transpose a matrix
point_inout_convex_non_degenerate_polygon	Test to evaluate if a point lies inside or strictly outside a polygon
quadratic_equation	To solve a quadratic equation
reference_system_change	Transformation of coordinates in a new reference system
three_plane_intersection	To assess the intersection between 3 planes
Vector_Product	To assess the cross product of two vectors
vector_rotation_axis_angle	To provide the rotation angle and axis of a 3D vector rotation
vector_rotation_Euler_angles	3D rotation of a given vector, provided the Euler angles.
vector_rotation_Rodrigues	3D rotation of a given vector, by means of Rodrigues formula.

Table 6.10
Program units (".f90") for the initial conditions (SPHERA, 2018, [1], folder "IC").

Program unit	Synthetic description
GeneratePart/SetParticles	To set particle initial positions
initialization_fixed_granular_particle	To initialize the most of the fixed SPH mixture particles (bed-load transport)
IsParticleInternal2/3D	To check whether a particle is internal to the 2/3D domain
SetParticleParameters	To set the initial particle quantities
SubCalcPreIdro	Hydrostatic pressure profiles (in case they are imposed as initial conditions)

Table 6.11
Program units (".f90") for the main algorithms (SPHERA, 2018, [1], folder "Main_algorithm").

Program unit	Synthetic description
Gest_Dealloc	Array deallocations
Gest_Trans	Introductory procedure for the main algorithm
Loop_Irre_2/3D	2/3D main algorithm
sphera	Main program unit

- time series of the main fluid dynamics variables (pressure and velocity) along the monitoring lines and points;
- flow rate hydrographs at the flow rate monitoring sections;
- hydrographs of the free surface height at the monitoring lines;
- 2D fields of the maximum values of the specific flow rate and the free surface height;
- time series of the interfaces of the model for dense granular flows;
- output files on the boundary treatment scheme DB-SPH;
- output files on the solid bodies.

6.2. SPHERA v.9.0.0: compilation and execution

SPHERA installation is straightforward, even because a set of executable files is already available as reference. SPHERA source and executable files are distributed on a dedicated Git repository [1] on GitHub. SPHERA executable files are released for Linux OS (compilers: both ifort and gfortran, with OpenMP libraries). The Makefile (under the folder "src") allows compiling SPHERA under different configurations, according to the assignation of the following variables: "EXECUTION" = "bin'" (optimized), "debug"; "COMPILATION_FLAGS" = "-O1" (optimized

Table 6.12
Fortran modules (".f90"; SPHERA, 2018, [1], folder "Modules").

Program unit	Synthetic description
Dynamic_allocation_module	Module to define the dynamically allocated variables
Hybrid_allocation_module	Module for derived types of both dynamically and statically allocated variables
I_O_diagnostic_module	To provide global interfaces to the subroutine "diagnostic"
I_O_file_module	Module for Input/Output file management
SA_SPH_module	Module for the semi-analytic approach (boundary treatment scheme)
Static_allocation_module	Module to define global and statically allocated variables
Time_module	Module for computational time recording

Table 6.13
Program units (".f90") for the neighbouring search, the smoothing operators and the interface detection (SPHERA, 2018, [1], folder "Neighbouring_Search").

Program unit	Synthetic description
CalcVarLength	Neighbouring search (pre-conditioned dynamic vector), SPH operators, interfaces.
CellIndices	To return the indices of the cell (positioning grid) a generic particle belongs to.
Particle/CellNumber	To return the ID of a generic positioning cell
CreaGrid	To create the background positioning grid
InterFix	Minor program unit
OrdGrid1	Ordering the numerical elements on the background positioning grid
w	To assess the kernel functions

Table 6.14
Program units (".f90") for post-processing (SPHERA, 2018, [1], folder "Post_processing").

Program unit	Synthetic description
calc_pelo	Post-processing to write the free surface height
CalcVarp	To calculate physical quantities at a monitoring point
cat_post_proc	To concatenate the ".txt" output files
CreateSectionPoints/s_ctime	Minor program units
electrical_substations	Substation-flooding damage model
GetVarPart	Getting particle values for monitoring
interface_post_processing	Post-processing the interfaces for bed-load transport phenomena
Memo_Ctl	Post-processing for monitoring lines and points
Memo_Results	To write detailed results for restart
Print_Results	Post-processing for the log file
result_converter	Post-processing for ".vtu" and ".vtk" files for Paraview
start_and_stop	Computational time recording
sub_Q_sections	To write the flow rate at the associated monitoring sections
Update_Zmax_at_grid_vert_columns	Current 2D fields of the water depth and specific flow rate
write_Granular_flows_interfaces	To print the interfaces for bed-load transport phenomena
write_h_max	2D fields of the maximum water depth and specific flow rate

execution), "-g -O0 -fbacktrace -C" (gfortran debug), "-g -O0 -traceback -C -check bounds -check noarg_temp_created -debug all" (ifort debug); "COMPILER" = "gfortran", "ifort"; "OMP_FLAG" = "-fopenmp" (gfortran omp); "-qopenmp" (ifort omp). Compilation is carried out by two consecutive command lines: "make touch"; "make".

The only mandatory argument in the command line for executing SPHERA is the name of the main input file (without the format extension ".inp").

6.3. Main input file of SPHERA v.9.0.0

The folder "input" of the code repository hosts a commented template of the main input file of SPHERA, where all the input parameters are defined, the meaning of their possible values is described and suggested, possible default values are reported. Table 6.18 presents a synthetic description of the contents of the main input file of SPHERA.

7. Modelling chain of SPHERA

An overview of the numerical modelling chain of SPHERA (Fig. 7.1) is herein discussed. The following procedure is generalized in Section 7.1 to deal with georeferenced domains.

Paraview (Kitware, [49]) is used to draw the geometrical features of the numerical domain of SPHERA and to save them in a ".ply" file. The numerical tool ply2SPHERA_perimeter (RSE SpA, [1]) converts the ".ply" file in two distinct output files. They have the same format as the sections "VERTICES" and "FACES" of the main input file of SPHERA. It is the vertices and faces of the solid surfaces within the numerical domain of SPHERA.

Table 6.15
Program units (".f90") for pre-processing (SPHERA, 2018, [1], folder "Pre_processing").

Program unit	Synthetic description
defcolpartzero	On the particle colours for visualization purposes
Diagnostic	Error message diagnostic
Gest_Input	Input check and management
Init_Arrays/ReadCheck/ReadInput…	Minor program units
ModifyFaces	To generate triangles from quadrilaterals
ReadBedLoadTransport	Reading input data for bed-load transport
ReadBodyDynamics	Reading input data for solid body transport
ReadDBSPH	Reading input data for the DB-SPH boundary treatment scheme
ReadInput	Reading input data
ReadInputBoundaries	Reading input data for the boundary treatment scheme SA-SPH
ReadInputControlLines	Reading monitoring lines
ReadInputControlPoints	Reading monitoring points
ReadInputControlSections	Reading control sections
ReadSectionFlowRate	Input management for the flow rate monitoring sections

Table 6.16
Program units (".f90") for the boundary treatment scheme SA-SPH (SPHERA, 2018, [1], folder "SA_SPH").

Program unit	Synthetic description
AddBoundaryContribution_to_CE/ME2/3D	To compute boundary terms for the 2/3D continuity/momentum equation
BoundaryMassForceMatrix2/3D	Generation of the generalized boundary mass force matrix in 2/3D
BoundaryPressureGradientMatrix3D	To generate the pressure gradient matrix
BoundaryReflectionMatrix2D	To generate the generalized reflection matrix R
BoundaryVolumeIntegrals2D	To compute the boundary volume integrals
ComputeBoundaryDataTab	To calculate the array to store close boundaries and integrals
ComputeKernelTable/…	Kernel-related parameters and integrals
ComputeSurfaceIntegral_WdS2D	Computing kernel surface integrals
ComputeVolumeIntegral_WdV2D	Computing kernel volume integrals
DefineBoundaryFace/SideGeometry3/2D	Definition of the boundary faces/sides in 3/2D
DefineBoundarySideRelativeAngles2D	Detection of the previous adjacent side and associated relative angle
DefineLocalSystemVersors	To define the directional cosines of the local reference system
FindBoundaryConvexEdges3D	To look for possible edges with an associated convex geometry
FindBoundaryIntersection2D	Intersection between a kernel support a boundary side
FindCloseBoundaryFaces3D/Sides2D	To find the "close" boundary faces/sides in 2/3D
GridCellBoundaryFacesIntersections3D	To find the boundary faces intercepted by the cells of the positioning grid
Interpolate/…	Interpolation of the boundary integrals
IWro2dro/J2Wro2/JdWsRn/WIntegr	To compute SA-SPH definite integrals
SelectCloseBoundarySides2D	Selecting the neighbouring boundary sides

Table 6.17
Program units (".f90") for time integration (SPHERA, 2018, [1], folder "Time_integration").

Program unit	Synthetic description
Euler	Explicit RK1 time integration scheme (Euler scheme)
Heun	Heun scheme: explicit RK2 time integration scheme.
time_step_duration	Computation of the time step duration according to stability constraints
stoptime	Stopping time for computational time recordings
time_integration	Explicit Runge–Kutta time integration schemes
time_integration_body_dynamics	Leapfrog time integration scheme for solid body transport

In case the boundary treatment method of Section 4 is used, SnappyHexMesh (OpenFOAM, OpenCFD Ltd, [50]) is used as a surface grid generator for the initial positioning grid of the DB-SPH elements.

Once the sections "VERTICES" and "FACES" are obtained from ply2SPHERA_perimeter and the input file for the positioning surface grid is produced by SnappyHexMesh, one completes the remaining sections of the main input file of SPHERA. This 3D CFD-SPH code is then executed.

The output files of SPHERA which contain the profiles (1D) of the fluid dynamics variables are visualized by means of Gnuplot (Williams & Kelley, [51]), which returns the output files in the ".eps" format. These files are read by GSView (Ghostgum Software Pty Ltd, [52]) and converted in the ".png" format. The profiles

Table 6.18
Sections and relevant quantities of the main input file of SPHERA v.9.0.0 [1].

Input file section	Synthetic description
Title	Test case title
Domain	Spatial resolution and choice of the boundary treatment scheme
Vertices	Vertices of the fluid domain boundaries
Lines/Faces	Vertex connections of the boundary lines/faces of the fluid domain in 2/3D
Boundaries	Features of the fluid domain bodies and boundaries (solid boundaries, open/inlet sections): initial conditions, boundary conditions, possible extrusions of water bodies from topography.
DBSPH	Quantities on the DB-SPH boundary treatment scheme, related to: spatial resolution at boundaries, MUSCL reconstruction scheme, geometry of semi-particles, slip conditions, limiters, monitors, number of files for the surface mesh (initial positions of the boundary elements), imposed kinematics (of the boundaries), inlet/outlet sections.
Bed-load transport	Input quantities for the following features: scheme for dense granular flows, erosion criterion, saturation scheme, monitors, liquefaction scheme.
Medium	Input physical quantities on the fluid and solid phase properties and the scheme for dense granular flows: bulk modulus, viscosity, saturation conditions, internal friction angle, limiting viscosity, maximum viscosity, effective porosity, mean diameter of the solid grains.
Body dynamics	Input physical quantities on the scheme for body transport in fluid flows: possible imposed kinematics; number of bodies; spatial resolution within the solid bodies; friction angle; limiters. For each body, the following quantities are requested: number of elements; mass; vectors of the initial position, velocity and angular velocity; tensor of the mass moment of inertia (if this is constant and not computed); initial orientation of the body with respect to the reference system. For each body element, the following quantities are requested: side lengths of the element; vector of the initial position; initial rotation of the element with respect to the reference system; Boolean operator to treat the element when configuring its reference body.
Run parameters	Final time, *CFL*, C_v and time integration scheme; weight of the partial smoothing; numerical quantities for memory management.
General physical properties	Gravity acceleration vector; reference pressure.
Restart	Frequency for writing the restart files
Output regulation/draw options	Frequency for writing SPHERA output files
Control points	Position of the monitoring points
Control lines	Position and discretization of the monitoring lines
Section flow rate	Geometry of the monitoring sections for the flow rate
Substations	Geometries of the electrical substations for the substation-flooding damage scheme and substation type (high-voltage transmission substation, medium-voltage distribution substation, low-voltage distribution substation). Each substation is described by a polygon.

simulated are compared with the analogous experimental (or numerical) profiles available from experimental images or the scientific literature (indexed journals; Open-Data archives). In this instance, it is normally admitted the digitization of experimental and numerical profiles from published sources by means of Engauge Digitizer (Mitchell et al. [53]), with proper citation of the source, in order to validate the code.

Starting from its 3D output, SPHERA also produces the 2D synthetic fields of the maximum fluid depth and the maximum specific flow rate (Section 6.1). These 2D output fields are post-processed by means of Grid Interpolator (RSE SpA, [1]).

Paraview shows the 2D and 3D fluid dynamics fields produced by SPHERA and returns the associated image files. These are concatenated in ".gif" animations by means of Image Magick (ImageMagick Studio LLC, [54]). The compression of these animations, necessary in case the concatenation involves many files, is carried out by means of Virtual Dub [55], which returns an ".avi" video output file.

The numerical modelling chain is based on free tools: FOSS, freeware or OpenData. All the items of the chain are FOSS, but the "Freeware" tools (GitHub for public repositories and GSView; they are simply free) and SRTM3, which is "Open-Data" (dataset available upon public and free access).

It is possible to support or replace the auxiliary tools of the modelling chain with more effective software, if available (e.g., commercial compilers; DEM files with finer spatial resolutions than SRTM3). The replacement of a free tool with a proprietary tool (with charge) is normally a reversible procedure which does not alter the functioning of the modelling chain.

7.1. Applications with georeferenced domains

In case of georeferenced domains, the beginning of the procedure of Section 7 is generalized as follows.

The dataset SRTM3 (USGS, 2014, [56]) provides a repository of DEM ("Digital Elevation Models") with an almost global cover and spatial resolution of 1" (in terms of geographical coordinates) or ca.31m (maximum/coarser spatial resolution in terms of cartographic coordinates). With respect to the former free DEM datasets (finest spatial resolution of 3"), SRTM3 has permitted

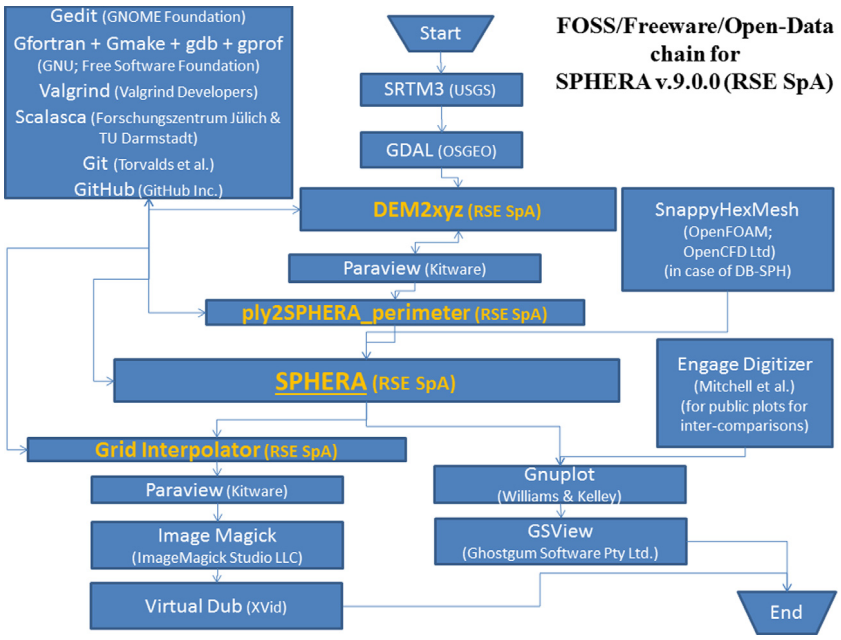

Fig. 7.1. Free tools of the numerical chain of SPHERA v.9.0.0.

a relevant improvement in using Open-Access DEM. The SRTM3 products are available in the format ".tif". Considering its global coverage, SRTM3 provides a root-mean-square error on heights of ca.6 m, even though the error kurtosis is very high (Rexer & Hirt, 2014, [57]). However, these estimations refer to almost the whole terrestrial surface, included the high-latitude regions where errors are definitely higher.

The software tool GDAL (OSGEO, [58]), the main QGIS library, can be used as an independent code. In the frame of the present modelling chain, GDAL allows to convert the DEM file format ".tif" of SRTM3 in the alternative format ".dem".

DEM2xyz (RSE SpA, [1]) reads the DEM file (".dem" format), converts the geographic coordinates in Cartesian coordinates over a regular grid and writes the resulting DEM on an output file (".xyz" format), possibly coarsening the spatial resolution.

Paraview reads the ".xyz" output file of DEM2xyz and elaborates a 2D Delaunay grid starting from the DEM vertices. Paraview also allows to cut the numerical domain (the cuts have to be far enough from the water bodies not to disable the procedures to extrude the water bodies from the DEM), circumscribes the water bodies, draws the possible filing/digging regions, detects the dam toe and the most upstream point over the coastline of the water bodies.

The above information, derived from Paraview, is transferred over the main input file of DEM2xyz, which is executed again to modify the already computed DEM, by reconstructing the bathymetry below the water bodies and the possible assignation of the digging/filling regions (uniform height within the same region), after a verification/modification on the normal vectors to the surface elements of the DEM. At this point, Paraview is used again to draw those geometrical figures which are necessary to initialize some variables in the main input file of SPHERA, in order to detect water bodies, earth-filled dams and monitoring elements.

The output elements of the procedure described above are synthesized in a ".ply" file by means of Paraview. Provided this file, the procedure of Section 7 can be followed.

8. Tutorials, validations and applications

SPHERA v.9.0.0 has been validated an applied on more than forty test cases, included the 34 tutorials of the code repository, whose input files are updated with the last code release. Each tutorial has up to seven configuration variants. Some of the test cases of SPHERA are described in the following sub-sections and refer to papers on International Journals.

The aim of this section is just to recall the code tutorials, and other validations and applications. However, the images shown here are unpublished. The references for details and validations on the single test cases are available in each of the following sub-sections, which are grouped according to the associated application fields: floods (Section 8.1); landslides and wave motion (Section 8.2); sediment removal from water bodies (Section 8.3); sloshing tanks (Section 8.4);. Finally, a performance analysis is discussed (Section 8.5).

8.1. Floods

The most applications of SPHERA refer to floods (Fig. 8.1). SPHERA has simulated floods on real and complex topographies (domain spatial coverage up to some hundreds of squared kilometres) with transport of solid bodies and bed-load transport.

8.1.1. 2D impact of a water liquid jet on a flat plate

A 2D water liquid jet impacts a solid flat plate. An exact analytical solution is available for validation purposes. This very simplified configuration is introductory to any fluid–boundary and fluid–body interaction occurring during floods, landslides and wave motion (Section 8.2), sediment removal from water bodies (Section 8.3) and sloshing tanks (sec.8.4). The seven variants of this tutorial (SPHERA v.9.0.0, 2018, [1], folder "jet_plate") refer to Amicarelli et al. (2013, [5]) and Amicarelli et al. (2015, [4]).

Hereafter is described an example of simplified use of the numerical chain of SPHERA, provided that most of the chain tools are not requested for such a simple test case. In order to

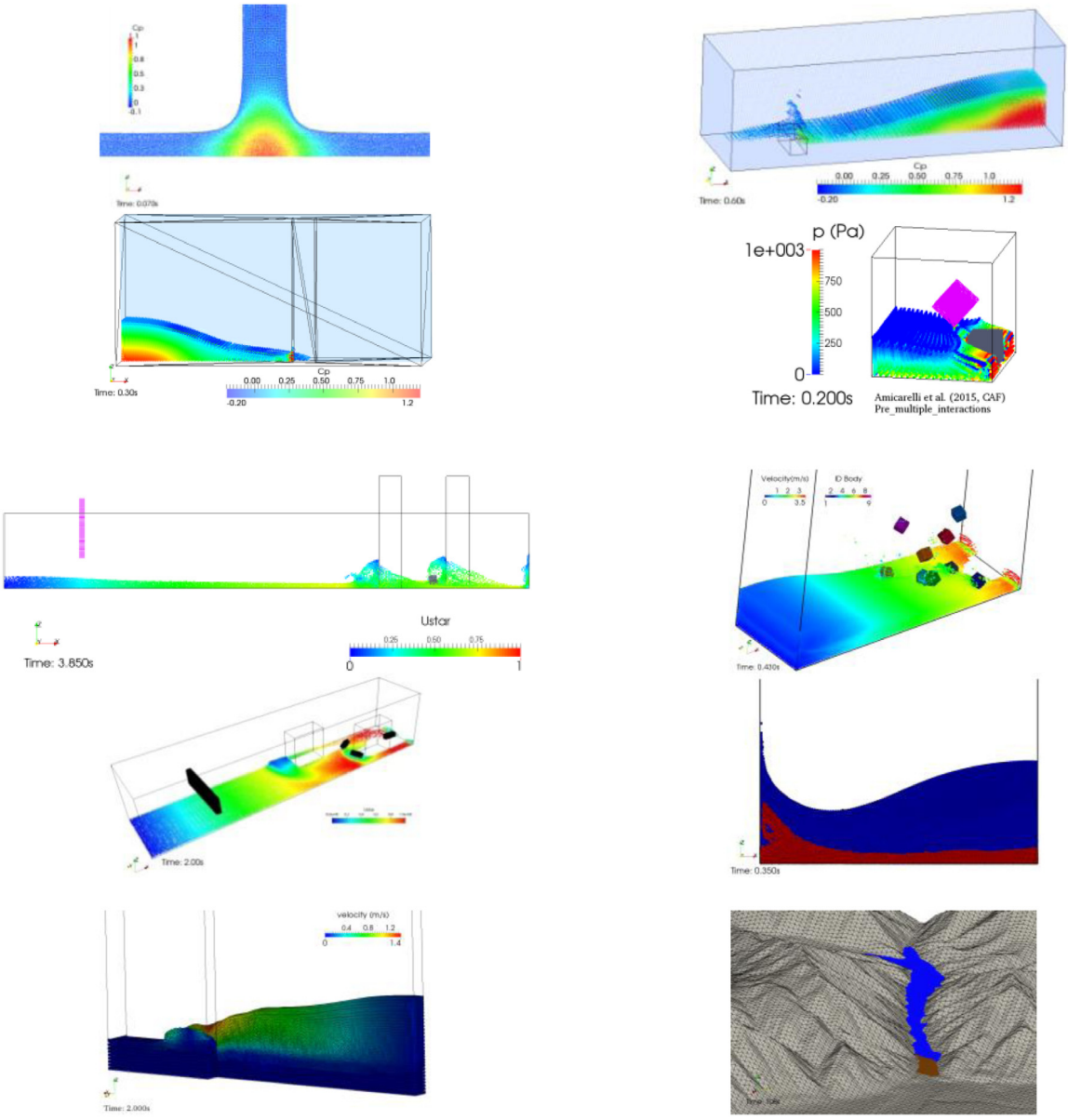

Fig. 8.1. Tutorials, validations and applications: floods. The test cases are shown from the left to the right panel, and from the top to the bottom row, following the section numbering. First row: 2D impact of a water liquid jet on a flat plate; impact of a dam-break flood front over a squat obstacle. Second row: impact of a dam-break flood front over a tall obstacle; impact of a dam-break flood front over two solid bodies. Third row: transport of a floating body during a dam-break flood with fixed obstacles; impact of a dam-break flood front over nine solid bodies. Fourth row: urban flood with the transport of large floating bodies; 2D erosional dam breaks. Fifth row: 3D laboratory erosional dam-break flood; erosional dam break on complex topography.

make the procedure simpler, one considers the variant n. 4 of this test case, where the semi-analytical approach (Section 2.1) replaces the DB-SPH boundary treatment (Section 4). One moves to the folder "~/SPHERA-v.9.0.0/input/release_v_9_0_0/24_ jet_plate/04_jet_plate_SASPH_low_res") on a Linux terminal. Considering an interactive serial job and that the source code can be also compiled with a GNU tool, the following command line is submitted "~/SPHERA-v.9.0.0/bin/SPHERA_v_9_0_0_ gfortran_bin.x jet_ plate_SASPH_low_res &". The following output files are generated and regularly updated during the execution, according to Section 6.1. The log file "jet_plate_SASPH_low_res.out" contains the synthetic information on the execution. SPHERA initially writes in the log file the input data and the name and size of the major allocated arrays. Every n time steps ($n = 100$ for this tutorial), the log file is updated with the maximum and minimum values (and the associated locations) of the main estimated quantities (pressure, velocity vector components and absolute value, …), the number of particles within the domain and entered/exited through the inlet/outlet sections since the beginning of the simulation, the time step duration and ID, the physical time, the percentage of fluid particles which need the assessment of the shear viscous term in the momentum equation. At the end of the simulation, SPHERA writes in the log file the list of arrays deallocated and the distribution of the elapsed time (12 m 14 s) among the main procedures of the code. The output file "VTKConverter_jet_plate_SASPH_low_res.pvd" collects (as a group) all the ".vtu" files of the fields of the output fluid dynamics quantities and can be very easily visualized by means of Paraview (Section 7), together with the output file representing the plate geometry "VTKConverter_jet_plate_SASPH_low_res_domain.vtk". Paraview is also used to save all the snapshots of the selected fields. These frames are concatenated by means of Image Magick

(Section 7) to produce animation files in ".gif" format. Virtual Dub (Section 7) converts the ".gif" files into compressed ".avi" animation files. The output file for the monitoring lines "monitoring_lines.txt" contains the time-dependent pressure profiles over the fluid–plate interface, as defined in the input file "jet_plate_SASPH_low_res.inp" (section "control lines"). As the final time is representative of stationary conditions, the last profile is copied and pasted on a blank file to be formatted as an input file for Gnuplot (Section 7). This file also contains the reference profile used for validation. This profile was digitized from Zoppè et al. (2004, [59]) by means of Engauge Digitizer (Section 7), following the Copyright license and providing a proper citation of the original source. Gnuplot is then used to save the two profiles (expressed in terms of non-dimensional quantities) on a ".eps" file, which is converted into a ".png" file (similar to Fig. 8.4) by means of GSView (Section 7).

8.1.2. Impact of a dam-break flood front over a squat obstacle

This tutorial (SPHERA v.9.0.0, 2018, [1], folder "db_squat_obstacle"; Amicarelli et al. 2013, [5]) permits to validate the code on a controlled laboratory test representing a 3D dam-break flood whose front impacts a squat obstacle.

8.1.3. Impact of a dam-break flood front over a tall obstacle

This tutorial (SPHERA v.9.0.0, 2018, [1], folder "db_tall_obstacle"; Amicarelli et al. 2013, [5]) permits to validate the code on a further laboratory test representing a 3D dam-break flood whose front impacts a tall obstacle.

8.1.4. Impact of a dam-break flood front over two solid bodies

This demonstrative tutorial (SPHERA v.9.0.0, 2018, [1], folder "db_2bodies"; Amicareli et al. 2015, [4]) represents a simplified dam break-flood front which impacts two solid bodies. Fluid–body, body–body and body–boundary interactions are solicited at a very coarse spatial resolution to demonstrate the absence of penetrations between the masses of the several media involved.

8.1.5. Transport of a floating body during a dam-break flood with fixed obstacles

This tutorial (SPHERA v.9.0.0, folder "db_body_exp_UniBas"; Amicarelli et al. 2015, [4]) simulates the transport of a floating solid body during a dam-break flood. The body interacts with the flood, two fixed obstacles and the solid boundaries of the domain. Validation is documented by comparing the numerical results with the available measures.

8.1.6. Impact of a dam-break flood front over nine solid bodies

This tutorial (SPHERA v.9.0.0, 2018, [1], folder "db_multi_body"; Amicarelli et al. 2015, [4]) simulates the impact of a simple dam-break flood over nine solid bodies (with different weights), juxtaposed at the beginning of the simulation to form a vertical plate. Multiple body–body interactions are simultaneous to fluid–boundary and body–boundary interactions.

8.1.7. Urban flood with the transport of large floating bodies

Albano et al. (2016, [60]) simulated a 3D complex configuration involving the multiple transport of rigid bodies in free surface flows. SPHERA was validated in a sequence of laboratory test cases carried out in Albano et al. (2016, [60]) on a rectangular tilting flume, which schematized the failure of a small dam affecting an urban floodplain near buildings and vehicles. The model showed a good reliability, in terms of water depth time evolution and time history of the body trajectories, demonstrating that it has reached a level of maturity that allows for quantitative comparison with complex experimental measurements; in this light, the model has demonstrated as a cost-efficient tool for urban flood analysis providing additional information that cannot be easily obtained from direct experimental observation such as water flow pressure and velocity.

8.1.8. 2D erosional dam breaks

An erosional dam break is represented by a dam-break flood propagating over a mobile bed. SPHERA demonstrative tutorial "edb_2D_demo" (SPHERA v.9.0.0) is represented in Fig. 8.1. Four other tutorials deal with validation and model inter-comparisons on erosional dam breaks (SPHERA v.9.0.0, 2018, [1], folders "edb_KarlSand", "edb_2D_FraCap02", "edb_2D_Spi05", "edb_2D_FraCap02_Taipei"; Amicarelli et al. 2017, [3]). These erosional dam breaks are also introductory to the more complex configurations of the following sub-sections.

8.1.9. 3D laboratory erosional dam-break flood

This erosional dam-break flood represents a tutorial (SPHERA v.9.0.0, 2018, [1], folder "edb_Pon10") which provides a code validation described in Amicarelli et al. (2017, [3]).

8.1.10. Erosional dam break on complex topography

This erosional dam-break flood represents a demonstrative tutorial (SPHERA v.9.0.0, 2018, [1], folder "edb_ICOLD"; Amicarelli et al. 2017, [3]) on complex topography.

8.2. Landslides and wave motion

SPHERA has been validated on a fast landslide in granular material (over complex topography) and on preliminary test cases for rock landslides (Fig. 8.2), also considering their interaction with water bodies and the associated wave motion.

8.2.1. 2D Vajont experiment — Padua hydraulic laboratory (1968)

This tutorial (SPHERA v.9.0.0, 2018, [1], folder "2D_Vajont_experiment"; Manenti et al. 2018, [61]) represents a characteristic cross-section of the Vajont artificial basin reproduced in the scale model of Padua (1968) for evaluating the effect of landslide falling velocity on the maximum wave run-up. This test allows evaluating the landslide-water coupled dynamics (with both stored and pore water) and code validation can be attained through the experimental measures.

8.2.2. Body–boundary and body–body impingements

These tutorials (SPHERA v.9.0.0, 2018, [1], folders "body_boundary_impacts" and "body_body_impacts"; Amicarelli et al. 2015, [4]) represent five configurations of body–boundary and body–body impingements, provide a code validation by comparison with analytical solutions and introduce to the applications on rock landslides with topography. These tutorials are also relevant for the transport of solid bodies during floods.

8.2.3. 2D free falls of solid wedges on still water

This tutorial (SPHERA v.9.0.0, 2018, [1], folder "wedge_falls_on_still_water"; Amicarelli et al. 2015, [4]) represents four 2D configurations of free falls of solid wedges on the free surface of a still water reservoir. This test case represents the first stage of the interaction between a falling rock and a water body and provides several code validations by comparisons with the available measures. This tutorial is also relevant for the transport of solid bodies during floods.

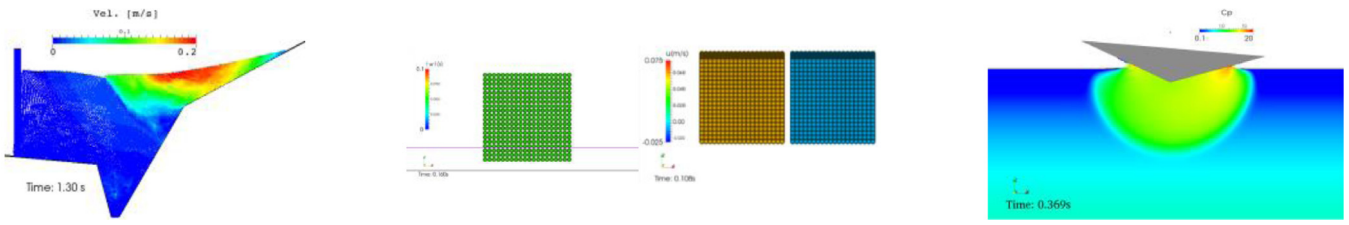

Fig. 8.2. Tutorials, validations and applications: landslides and wave motion. The test cases are shown from the left to the right panel, following the section numbering. 2D Vajont experiment (landslide and tsunami); body–boundary and body–body impingements; 2D free falls of solid wedges on still water.

8.3. Sediment removal from water bodies

A 2D laboratory test reproduces a small-scale idealized flushing manoeuvre induced by the opening of the dam bottom outlet in a long, narrow artificial basin. The coupled water–sediment dynamics can be suitably simulated (Manenti et al. 2012, [6]), as shown by the comparison between experimental and calculated final sediment profile (Fig. 8.3, left panel).

8.4. Sloshing tanks

SPHERA has been validated on two 2D sloshing tanks (Fig. 8.3, right panel). These test cases are introductory to the application fields of fuel sloshing tanks and seismic dampers. These tutorials (SPHERA v.9.0.0, "sloshing_tank_TbyTn_1_07" and "sloshing_tank_TbyTn_0_78"; Amicarelli et al. 2013, [5]) permitted to validate the code on the measures of 2D laboratory sloshing tanks with resonance or beats (the period of the force imposed to the tank is 1.07 or 0.78 times the sloshing natural period of the tanks, which are partially filled with water).

8.5. Examples of performance analysis

8.5.1. Validations

SPHERA has been validated on several test cases, reported on original research papers of International Peer-Reviewed Journals (Di Monaco et al. 2011, [7]; Manenti et al. 2012, [6]; Amicarelli et al. 2013, [5]; Amicarelli et al. 2015, [4]; Albano et al. 2016, [60]; Amicarelli et al. 2017, [3], Manenti et al. 2018, [61]; Paggi et al. 2019, [62]). Hereafter, a couple of examples of the validation of the release SPHERA v.9.0.0 are shown.

The first example considers the 2D water jet impact over a flat plate (Fig. 8.2, top left panel). The jet diameter is $D = 0.028$ m and the absolute value of the undisturbed velocity is $U = 19.61$ m/s. The spatial resolution is defined by $dx = 5.0 \times 10^{-4}$ m, $dx_s = 0.5dx$ and $h/dx = 1.3$. Euler scheme is used for time integration with $CFL = 0.025$. The boundary treatment scheme DB-SPH (Section 4) is applied. The boundary conditions are imposed assigning the inlet velocity to the incoming particles and a null pressure in a buffer zone at the lateral outlet sections. Fig. 8.4 (left panel) shows the validation of the pressure coefficient profile over the plate, by comparison with the available analytical solution (Zoppè, 2004, [59]).

The second example (symmetric variant of the right panel of Fig. 8.2) considers an experimental free fall of a solid wedge on still water (the symmetric fall of a light wedge reported in Hay et al. 2006, [63]). The effects of air on the solid wedge are neglected (mono-phase simulation). This hypothesis introduces some discrepancies with respect to the experiments, mainly in terms of initial conditions (i.e. at the impact time). The base of the triangular wedge is $L = 0.610$ m, its height is $H = 0.111$ m. The origin of the reference system is located where the lowest point of the triangle impacts the free surface. The spatial resolution is defined by $dx = 0.0013$ m, $dx_s = 0.5dx$ and $h/dx = 1.3$. Leapfrog scheme is used for time integration with $CFL = 0.1$. The experimental initial conditions of the wedge fall (before the impact time) consider that the body velocity is null when the height of the triangle barycentre is $z_{CM} = 0.721$ m. The body specific mass is $m_B = 50.000$ kg/m (2D configuration). Fig. 8.4 (right panel) compares the time series of the simulated acceleration (normalized on g) with the available measures and the estimations provided by the asymptotic method (Hay et al. 2006, [63]). At the impact time, only gravity acts on the body. As the wedge crosses the free surface, the hydrodynamic thrust grows until a maximum (t = ca.0.01 s) and then exponentially decreases. The numerical results show a good agreement with the experimental values, in terms of time evolution, peak value and absolute errors. At the same time, the numerical evolution is smoother than the experimental one due to the solid deformation of the experimental body (the numerical body is rigid) causing small-scale pressure oscillations in the fluid domain.

Further, the initial conditions are not exactly the same, as reported above.

8.5.2. Optimization and parallelization

SPHERA HPC simulations typically involve some millions of numerical particles. SPHERA is parallelized by means of the OpenMP technique, apart from a residual part of the code which is not parallelized at all. This Non-Parallelized code Fraction (*NPF*) represents ca.5% of the code in terms of elapsed time of serial executions.

Among the most advanced HPC applications of SPHERA, one notices 3D floods on full-scale complex topographies with a domain spatial coverage up to some hundreds of squared kilometres (e.g., Amicarelli et al. 2017, [3]) and one of the first CFD applications on slider bearings with 3D complex surfaces (Paggi et al. 2019, [62]).

SPHERA adopts peculiar optimization schemes and procedures such as the pre-conditioned dynamic vector technique for the SPH neighbouring search, an erosion criterion (where applicable, Manenti et al. 2012, [6]) and the limiting viscosity (Manenti et al. 2018, [61]) for the scheme on dense granular flows, among the others.

As an example of SPHERA HPC applications, the variant n.1 of the tutorial n.34 (SPHERA, 2019, [1]) is here considered for a performance analysis. This describes the symmetric free fall of a solid wedge on still water (Amicarelli et al. 2015, [4]). With respect to the release SPHERA v.9.0.0, the simulation here discussed refers to a finer spatial resolution ($dx = 0.0013$ m) and no-slip conditions (SPHERA git commit of 9 January 2019; compute node with 2×18-core Intel Xeon E5-2697 v4 Broadwell processors at 2.30 GHz). 1.8 million SPH particles are used (1′703′858 fluid particles, 80′722 solid body particles) for 6'978 time steps. The elapsed time of the serial simulation is distributed among the following procedures: search for neighbouring particles (59%), momentum balance equations (16%), fluid–body interactions and

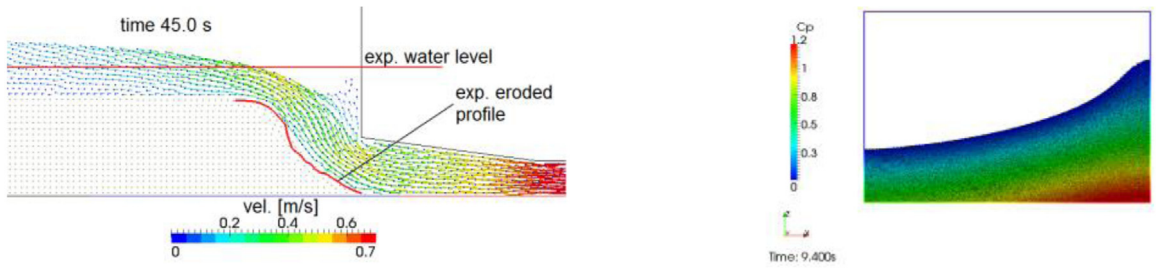

Fig. 8.3. Tutorials, validations and applications. 2D sediment flushing (left panel); sloshing tank under resonance (right panel).

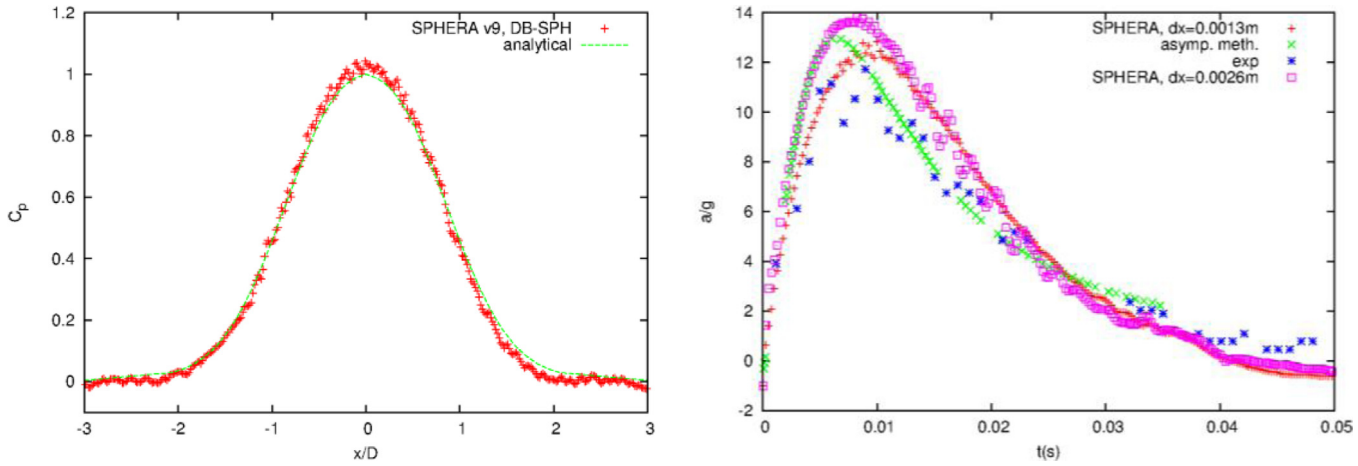

Fig. 8.4. Examples of validations of SPHERA v.9.0.0. Left panel. 2D impact of a water liquid jet on a flat plate (tutorial n.24, variant n.2): pressure coefficient profile under stationary conditions (numerical results and reference analytical solution from Zoppè (2004, [59])). Right panel. Symmetric free fall of a solid wedge on still water (tutorial n.34, variant n.1; git commit of 9 January 2019; no slip conditions; $dx = 0.0013$ m): time series of the normalized vertical acceleration (further comparisons with the asymptotic method and an analogous SPHERA simulation with a coarser spatial resolution).

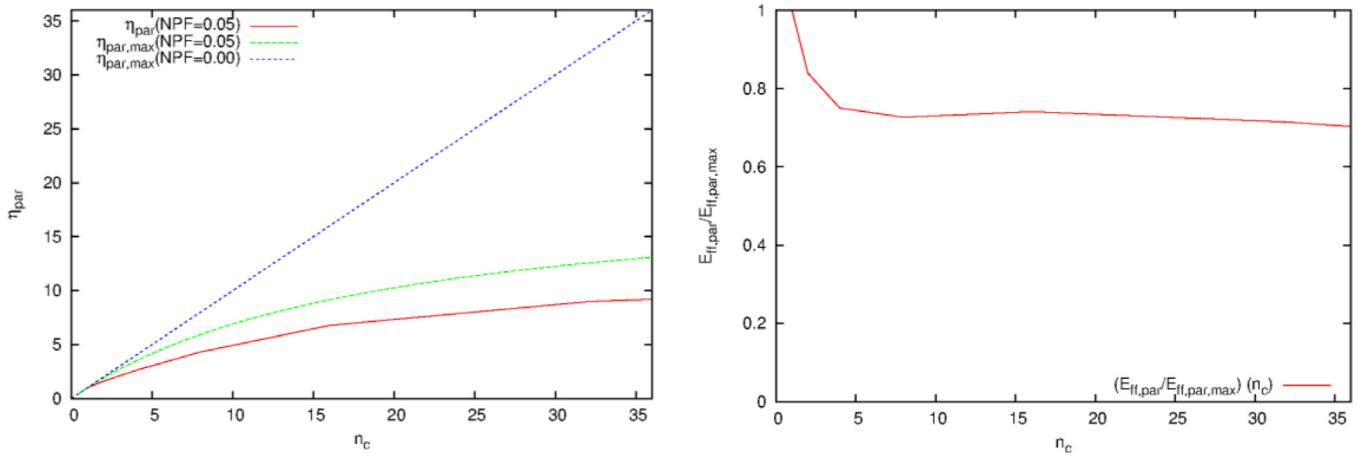

Fig. 8.5. Symmetric free fall of a solid wedge on still water (Amicarelli et al. 2015). Simulation with SPHERA v.9.0.0 (tutorial n.34, variant n.1; git commit of 9 January 2019; no slip conditions; $dx = 0.0013$ m, compute node with 2×18-core Intel Xeon E5-2697 v4 Broadwell processors at 2.30 GHz; Non-Parallelized Fraction of the code NPF = ca.0.05; 1.8 million SPH particles; 6′978 time steps; elapsed time of 4h41′46″ for the 36-core run). Left panel. Parallelization speed-up (η_{par}) and its maximum value (provided $NPF = 0.05$) depending on the number of cores n_c. Right panel. Relative parallelization efficiency (i.e. ratio between the parallelization efficiency and its maximum ideal value) depending on the number of cores n_c.

body dynamics (6%), continuity equation (6%), other procedures (13%; fluid and body particle reordering on the background grid, partial smoothing of velocity, fluid–boundary interactions, partial smoothing of pressure, time integration for the particle positions, pre-processing, post-processing, ICs, constitutive equation, etc.). The elapsed time for the 36-core run is 4h41′46″ (16906 s).

Under $NPF > 0$, the maximum parallelization efficiency $E_{ff,par,\max} = [n_c NPF + (1 - NPF)]^{-1}$ is smaller than the unity and depends on the number of cores n_c (here $NPF = 0.05$). Under the same conditions, the maximum parallelization speed-up $\eta_{par,\max} = \left[1 + (1 - NPF)\left(\frac{1}{n_c} - 1\right)\right]^{-1}$ is smaller than n_c.

Fig. 8.5 (left panel) reports the parallelization speed-up η_{par} and its maximum/ideal value $\eta_{par,max}$ depending on the number of cores n_c. Fig. 8.5 (right panel) quantifies the relative parallelization efficiency, i.e. the ratio between the parallelization efficiency and its maximum ideal value as function of the number of cores.

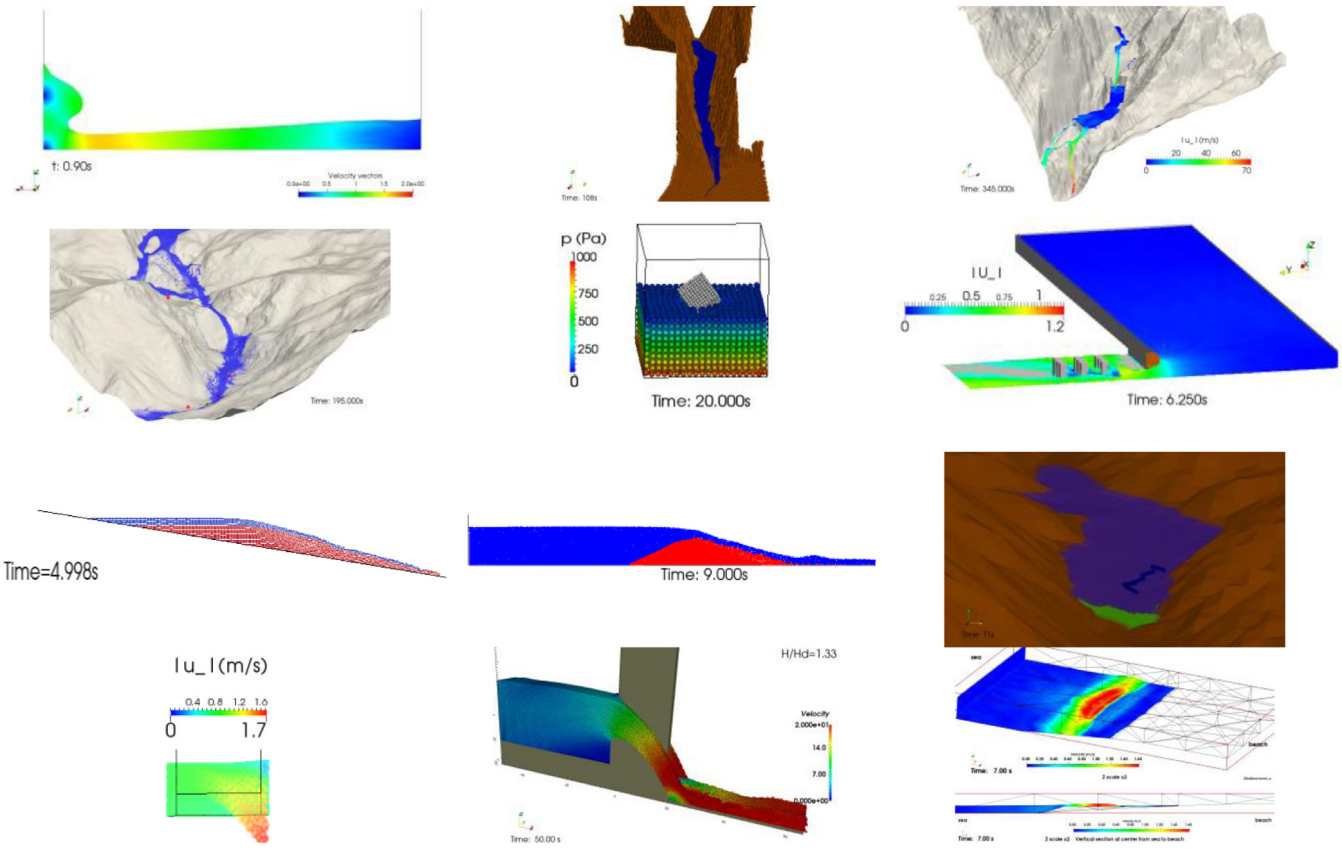

Fig. 9.1. Further applications: floods. The test cases are shown from the left to the right panel, and from the top to the bottom row, following the section numbering. First row: impact of a 2D dam-break frontal wave on a solid vertical wall; dam break on complex topography; dam-break scenarios for the concrete dam of Alpe Gera. Second row: propagation of the Alpe Gera dam-break flood on the residential areas and the electrical substations of the municipality of Lanzada; stability of a floating cube; urban dam-break flood scenarios. Third row; erosional dam break on a 2D slope; dike overtopping; dam-breach flood with transport of tree trunks. Fourth row: sub-critical flow passing by a rectangular side weir; spillway with pier (USACE benchmark); tsunami (ISEC benchmark n.1).

SPHERA has being under assessment to improve its parallelization and optimization strategy considering the improvement of its OpenMP parallelization (included the zeroing of *NPF*), a wider use of the compilation options, the code vectorization and the future use of GPU cards (beyond the CPUs).

9. Further applications

Further applications of SPHERA v.9.0.0 refer to studies in progress (or to minor publications) and are described in the following sub-sections, which are grouped according to the associated application fields: floods (Section 9.1); landslides and wave motion (Section 9.2); sediment removal from water bodies (Section 9.3); hydrodynamic lubrication (Section 9.4). No validation is yet available for the following test cases.

9.1. Floods

On-going applications of SPHERA introduce new applications such as the assessment of flood-control works and the flood-induced damage related to the functioning of the electrical substations (Fig. 9.1).

9.1.1. Impact of a 2D dam-break frontal wave on a solid vertical wall

Two advanced boundary treatment techniques of SPHERA are in the process of being further validated on 2D dam break flows in laboratory experiments by Lobosvky et al. [64], numerically reconstructing the time trend of the primary wave pressures acting on the solid vertical wall downstream and the primary and secondary wave height at specific positions along the channel (Albano et al., submitted). The first employed boundary treatment technique (Section 2.1) is based on the computation of volume integrals within the truncated portions of the kernel supports at boundaries; the second one (Section 2.3) stems from the extension of the ghost-particle boundary method for mobile boundaries adapted to free-slip conditions.

9.1.2. Dam break on complex topography

This tutorial (SPHERA v.9.0.0, 2018, [1], folder "db_ICOLD") provides a demonstrative study of an instantaneous and complete break of a concrete dam whose flood propagates on a complex and full-scale topography.

9.1.3. Dam-break scenarios for the concrete dam of Alpe Gera

This tutorial (SPHERA v.9.0.0, 2018, [1], folder "db_Alpe_Gera") represents the full-scale representation of three dam-break scenarios involving the highest Italian working concrete dam (Alpe Gera) and the downstream concrete dam of Campo Moro. Flood propagates on real topography and a flood-control work is simulated.

9.1.4. Propagation of the Alpe Gera dam-break flood on the residential areas and the electrical substations of the municipality of lanzada

This tutorial (Amicarelli et al., submitted [65]) simulates the follow-up of one of the scenarios of Section 9.1.3 over a downstream domain involving four residential areas and two electrical substations of the municipality of Lanzada (Sondrio, Italy), whose

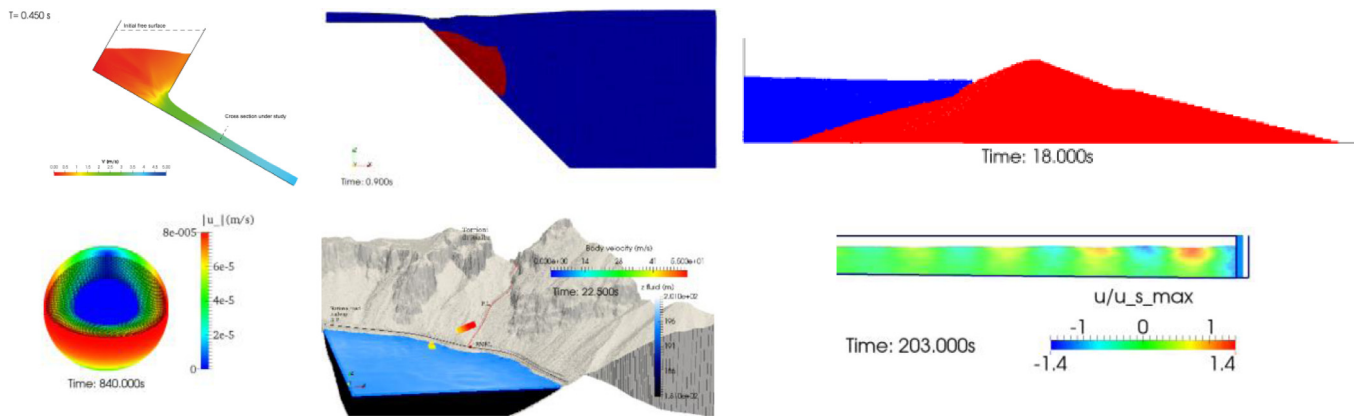

Fig. 9.2. Further applications: landslides and wave motion. The test cases are shown from the left to the right panel, and from the top to the bottom row, following the section numbering. First row: laboratory dry granular flows; 2D submerged landslide; liquefaction of San Fernando Lower dam. Second row: spherical Couette flows; Torrioni di Rialba rock-toppling runout and landslide-water impact simulation; swell sea wave.

flooding-related damage is assessed in terms of quantities associated with substation out-of-service and costs to restore the electrical components of the substations.

9.1.5. Stability of a floating cube

This simple and demonstrative tutorial (SPHERA v.9.0.0, 2018, [1], folder "floating_cube_stability") represents a solid cube leaned on still water at a rough spatial resolution. This test is introductory to more complex applications dealing with the transport of solid bodies within fluid flows.

9.1.6. Urban dam-break flood scenarios

This tutorial (Amicarelli et al., submitted [65]) represents the SPH simulation of two scenarios of a laboratory urban dam-break flood for validation purposes.

9.1.7. Erosional dam break on a 2D slope

A 2D collapse of granular–liquid mixture over rigid inclined flatbed is represented (Ziane, Khellaf, Amicarelli; in progress). The validation is documented by comparing the numerical front position of the mixture with the available measures (Berzi et al. 2012, [66]).

9.1.8. Dyke overtopping

This tutorial (SPHERA v.9.0.0, 2018, [1], folder "dike_breach _2D_expSchHag12JHR_ID22") represents the SPH simulation of a laboratory dike overtopping to validate the code on a riverine flood.

9.1.9. Dam-breach flood with transport of tree trunks

This tutorial (SPHERA v.9.0.0, 2018, [1], folder "dam_breach_ ICOLD_trunks") represents a flood, which is induced by a dam breach (i.e. break of an earth-filled dam by the propagation of a breach channel), propagates on complex topography and transports five solid trunks. This study simultaneously involves the transport of granular material and solid bodies during a flood event.

9.1.10. Sub-critical flow passing by a rectangular side weir

This tutorial (SPHERA v.9.0.0, 2018, [1], folder "rectangular _side_weir_Fr_0_491") reproduces a laboratory experiment on the stationary regime of a sub-critical free-surface flow ($Fr = 0.491$) passing by a rectangular side weir.

9.1.11. Spillway with pier (USACE benchmark)

This tutorial (SPHERA v.9.0.0, 2018, [1], folder "spillway _withpiers") represents the SPH simulation of a spillway with pier under rising flood levels. The laboratory test of uncontrolled overflow spillway crest (with pier) are reproduced for validation purposes.

9.1.12. Tsunami (ISEC benchmark n.1)

This tutorial (SPHERA v.9.0.0, 2018, [1], folder "tsunami _benchmark1_ISEC") represents the SPH simulation of transient wave propagation of a tsunami wave attack toward beach. The results are compared with the experiment for validation purposes.

9.2. Landslides and wave motion

SPHERA has represented several fast landslides in rocks or granular material (Fig. 9.2), in terms of laboratory and full-scale configurations on real topographies, their possible interaction with water bodies and the associated wave motion.

9.2.1. Laboratory dry granular flows

Following the work presented in [67,68], G. Viccione, B. Tagliafierro and L. Sarno (in progress) explore the capability of SPHERA to numerically reproduce dry granular flows which kinematics is depending both on the collisional and frictional momentum exchanges [69]. Results show the development of a hypercritical fluid flow (being the Froude number $F_r >> 1$) at the cross section under investigation.

9.2.2. 2D submerged landslide

This tutorial (SPHERA v.9.0.0, 2018, [1], folder "submerged_ landslide") represents a laboratory submerged landslide triggering a tsunami wave. This study aims to validate the code on a fast landslide and its interaction with a water reservoir.

9.2.3. Liquefaction of San Fernando Lower dam

This tutorial (SPHERA v.9.0.0, 2018, [1], folder "San_Fernando _Lower_ _dam_liquefaction") represents the real full-scale liquefaction of San Fernando Lower dam (USA, 1971) triggered by an earthquake of $M = 6.6$. This study aims to validate the scheme for liquefaction and apply the code on a complex landslide which interacts with a hydroelectric reservoir.

9.2.4. Spherical Couette flows

This tutorial (SPHERA v.9.0.0, 2018, [1], folder "spherical _Couette_flows") represents three configurations of 3D spherical Couette flows (i.e. a fluid shell moving between two rotating solid spheres). Thanks to analytical solutions available, this test case aims to validate the code under laminar regimes (commonly recorded in dense granular flows), the treatment of mobile boundaries and the propagation of the shear stress waves induced. These features are also introductory to represent earthquake-induced landslides.

9.2.5. Torrioni di Rialba rock-toppling runout and landslide-water impact simulation

This test case (Longoni et al. in progress) deals with the 3D modelling of a rock-topple landslide. The proposed case study is focused on Torrioni di Rialba: 135-metre high rock cliffs that are facing on Como Lake. They are subjected to instability problems due to their geological setting as described in Arosio et al. (2019, [70]) and Brambilla et al. [71]. SPHERA simulates different landslide collapse scenarios and runouts. Simulation starts from the rock impact with the terrain, goes on with its sliding along the 300 m slope below it, continues with its impact with Como Lake and it ends with the surface wave propagation. This test case is a first example of the use of a numerical method to represent a toppling event in 3D, including the run-out phase and the interaction with a water body.

9.2.6. Swell sea wave

This tutorial (SPHERA v.9.0.0, 2018, [1], folder "wave_motion _for_WaveSAX") represents an introductory and demonstrative test case on the generation, propagation and dissipation of a swell sea wave over 3D inclined and simplified bathymetry.

9.3. Sediment removal from water bodies

SPHERA has simulated two demonstrative processes on sediment removal from a water reservoir by means of the opening of a discharge channel (i.e. flushing; Fig. 9.3, left panel). The associated tutorials (SPHERA v.9.0.0, 2018, [1], folders "flushing_2D" and "flushing_3D") analyse 2D and 3D demonstrative and simplified configurations to assess different erosion criteria and further validate the scheme for dense granular flows.

9.4. Viscous flow modelling for hydrodynamic lubrication problems

A further development of SPHERA to treat hydrodynamic lubrication between bodies, considering a viscous fluid trapped between a rigid surface and a slider, is under investigation (Paggi et al. in progress). Historically, this topic is of great interest in tribology, where lubrication is used to reduce frictional resistances to relative motion of machine elements. SPHERA predictions have been validated in relation to the simulation of thin-film viscous fluid flow in slider bearings with parallel or linearly diverging profiles, where closed-form solutions are available, see Almqvist et al. (2014, [72]) for an overview of lubrication problems and Paggi & He (2015, [73]) and Paggi & Ciavarella (2010, [74]) for the analysis of the relative motion between two rough surfaces. In this regard, SPHERA is opening new perspectives for modelling and simulation of complex hydrodynamic lubrication problems involving complex textured or rough surfaces, possibly bio-inspired by nature, for which no closed-form solutions are available to characterize their performance.

10. Conclusions

SPHERA v.9.0.0 (RSE SpA) is a CFD-SPH research code featured by several numerical schemes dealing with: transport of solid bodies in fluid flows; treatment of fixed and mobile solid boundaries; dense granular flows and erosion criteria. SPHERA major publications involve the following application fields: floods with transport of solid bodies and bed-load transport; fast landslides and their interactions with water reservoirs; sediment removal from water bodies; fuel sloshing tanks; viscous flow modelling for hydrodynamic lubrication problems. SPHERA is developed and distributed on a GitHub public repository (SPHERA, 2018, [1]) with more than thirty tutorials, thus allowing the code availability and possible modification, and the repeatability of the published test cases. The whole numerical chain of SPHERA is free.

CRediT authorship contribution statement

Andrea Amicarelli: Conceptualization, Methodology, Software (testing of existing code components), Validation, Formal analysis, Writing - original draft, Writing - review & editing, Visualization. Sauro Manenti: Methodology, Software (testing of existing code components), Validation, Writing - original draft, Writing - review & editing, Visualization. Raffaele Albano: Software (testing of existing code components), Validation, Writing - original draft, Writing - review & editing, Visualization. Giordano Agate: Software (testing of existing code components), Writing - original draft, Writing - review & editing, Visualization. Marco Paggi: Software (testing of existing code components), Validation, Writing - original draft, Writing - review & editing, Visualization. Laura Longoni: Software (testing of existing code components), Writing - original draft, Writing - review & editing, Visualization. Domenica Mirauda: Software (testing of existing code components), Writing - original draft, Writing - review & editing, Visualization. Latifa Ziane: Software (testing of existing code components), Writing - original draft, Writing - review & editing, Visualization. Giacomo Viccione: Software (testing of existing code components), Writing - original draft, Writing - review & editing, Visualization. Sara Todeschini: Writing - original draft, Writing - review & editing, Visualization. Aurelia Sole: Writing - original draft, Writing - review & editing, Visualization. Lara Martina Baldini: Software (testing of existing code components), Writing - original draft, Writing - review & editing, Visualization. Davide Brambilla: Writing - original draft, Writing - review & editing, Visualization. Monica Papini: Writing - original draft, Writing - review & editing, Visualization. Mohamed Cherif Khellaf: Writing - original draft, Writing - review & editing, Visualization. Bonaventura Tagliafierro: Software (testing of existing code components), Writing - original draft, Writing - review & editing, Visualization. Luca Sarno: Writing - original draft, Writing - review & editing, Visualization. Guido Pirovano: Writing - original draft, Writing - review & editing, Supervision (overall supervision of the software paper).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

SPHERA has been financed by the Research Fund for the Italian Electrical System (for "Ricerca di Sistema -RdS-"), at different stages:

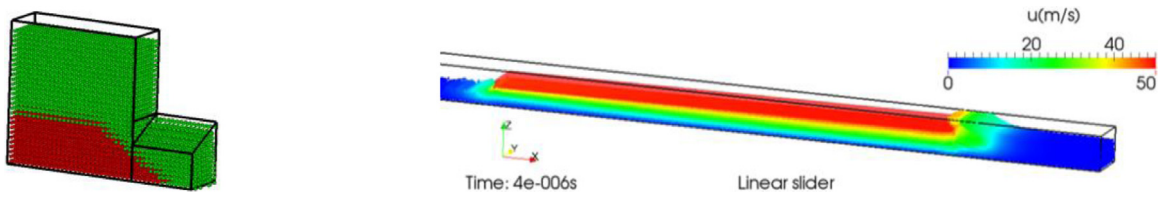

Fig. 9.3. Further applications: demonstrative test cases on sediment removal from water bodies (left panel); linear slider bearing (right panel).

- under the second period of RdS (2003–2005), where CESI SpA was the only beneficiary of the Research Fund for the Italian Electrical System;
- under the Contract Agreement between CESI Ricerca SpA and the Italian Ministry of Economic Development for the of RdS period 2006–2008, in compliance with the Decree of 8 March 2006;
- under the Contract Agreement between ERSE and the Ministry of Economic Development-General Directorate for Energy and Mining Resources (for the of RdS period 2009–2011) stipulated on 29 July 2009 in compliance with the Decree of 19 March 2009;
- under the Contract Agreement between RSE SpA and the Italian Ministry of Economic Development for the of RdS period 2012–2014, in compliance with the Decree of November 9, 2012;
- under the Contract Agreement between RSE SpA and the Italian Ministry of Economic Development for the RdS period 2015–2017, in compliance with the Decree of 21 April 2016. Reference project: 'A.5 - Sicurezza e vulnerabilità del sistema elettrico', Frigerio A. et al. 2015–2018.

"We acknowledge the CINECA award under the ISCRA initiative, for the availability of High Performance Computing resources and support". In fact, SPHERA validation has also been financed by means of the following instrumental funding HPC projects: HPCNHLW2, HSPHER9b, HPCNHLW1, HSPHCS9, HSPHERA9, HPCEFM18, FLR-RMPV, TNMRA01, HPCEFM7b, HPCEFM17, NMTFEPRA, HPCEFM16, HPCEFM15, HSPHMI14.

The release of the FOSS versions of SPHERA has been supported and promoted by the RSE Department Director Michele de Nigris, the RSE Deputy Director and Project Manager Antonella Frigerio, and the RSE Research Team Managers Guido Pirovano (since 2016) and Massimo Meghella (2015–2016).

SPHERA has been developed for RSE SpA by the following authors (list of the code authors, SPHERA, 2018, [1]): Andrea Amicarelli, Antonio Di Monaco, Sauro Manenti, Elia Giuseppe Bon, Daria Gatti, Giordano Agate, Stefano Falappi, Barbara Flamini, Roberto Guandalini, David Zuccalà, Qiao Cheng. SPHERA is free software released under the GNU General Public License (Free Software Foundation) and its Copyright is registered at SIAE. SPHERA is indexed by SPHERIC [75].

References

[1] SPHERA (RSE SpA), 2019, https://github.com/AndreaAmicarelliRSE/SPHERA, (Last Accessed 28 May 2019).
[2] Free Software Foundation, http://www.fsf.org/.
[3] A. Amicarelli, B. Kocak, S. Sibilla, J. Grabe, Int. J. Comput. Fluid Dyn. 31 (10) (2017) 413–434, http://dx.doi.org/10.1080/10618562.2017.1422731, JCR Impact Factor: 0.983.
[4] A. Amicarelli, R. Albano, D. Mirauda, G. Agate, A. Sole, R. Guandalini, Comput. & Fluids 116 (2015) 205–228, http://dx.doi.org/10.1016/j.compfluid.2015.04.018.
[5] A. Amicarelli, G. Agate, R. Guandalini, Internat. J. Numer. Methods Engrg. 95 (2013) 419–450, http://dx.doi.org/10.1002/nme.4514.
[6] S. Manenti, S. Sibilla, M. Gallati, G. Agate, R. Guandalini, J. Hydraul. Eng. ASCE 138 (3) (2012) 272–284.
[7] A. Di Monaco, S. Manenti, M. Gallati, S. Sibilla, G. Agate, R. Guandalini, Eng. Appl. Comput. Fluid Mech. 5 (1) (2011) 1–15.
[8] J.J. Monaghan, Rep. Progr. Phys. 68 (2005) 1703–1759.
[9] S. Adami, X.Y. Hu, N.A. Adams, J. Comput. Phys. 231 (2012) 7057–7075.
[10] M.R. Hashemi, R. Fatehi, M.T. Manzari, Int. J. Non-Linear Mech. 47 (2012) 626–638.
[11] F. Macia, L.M. Gonzalez, J.L. Cercos-Pita, A. Souto-Iglesias, Progr. Theoret. Phys. 128 (3) (2012) 439–462.
[12] A. Mayrhofer, B.D. Rogers, D. Violeau, M. Ferrand, Comput. Phys. Comm. 184 (2013) 2515–2527.
[13] M. Ferrand, D.R. Laurence, B.D. Rogers, D. Violeau, C. Kassiotis, Internat. J. Numer. Methods Fluids 71 (4) (2013) 446–472.
[14] R. Vacondio, B.D. Rogers, P.K. Stansby, P. Mignosa, Comput. Methods Appl. Mech. Engrg. 300 (442–460) (2016) http://dx.doi.org/10.1016/j.cma.2015.11.021.
[15] M. Gomez-Gesteira, B.D. Rogers, R.A. Dalrymple, A.J.C. Crespo, J. Hydraul. Res. 48 (Extra Issue) (2010) 6–27, http://dx.doi.org/10.3826/jhr.2010.0012.
[16] D. Le Touzé, A. Colagrossi, G. Colicchio, M. Greco, Internat. J. Numer. Methods Fluids 73 (2013) 660–691, http://dx.doi.org/10.1002/fld.3819.
[17] M.S. Shadloo, G. Oger, D. Le Touzé, Comput. & Fluids 136 (2016) 11–34.
[18] D. Violeau, B.D. Rogers, J. Hydraul. Res. 54 (1) (2016) 1–26.
[19] H. Gotoh, A. Khayyer, Coast. Eng. J. 60 (1) (2018) 79–103, http://dx.doi.org/10.1080/21664250.2018.1436243.
[20] S. Manenti, D. Wang, J.M. Domínguez, D. Li, A. Amicarelli, R. Albano, Water 11 (1875) (2019) 1–26.
[21] S. Gu, X. Zheng, L. Ren, H. Xie, Y. Huang, J. Wei, S. Shao, Water 9 (6) (2017) 387, (Switzerland).
[22] R. Vacondio, B.D. Rogers, P. Stansby, P. Mignosa, J. Hydraul. Eng. 138 (6) (2012) 530–541.
[23] A.J. Crespo, M. Gómez-Gesteira, R.A. Dalrymple, J. Waterw. Port Coast. Ocean Eng. 134 (6) (2008) 313–320, http://dx.doi.org/10.1061/(ASCE)0733-950X(2008)134:6(313).
[24] A. Khayyer, H. Gotoh, J. Hydraul. Res. 48 (2) (2010) 238–249, http://dx.doi.org/10.1080/00221681003726361.
[25] R. Vacondio, A. Dal Palù, P. Mignosa, Environ. Model. Softw. 57 (2014) 60–75, http://dx.doi.org/10.1016/j.envsoft.2014.02.003.
[26] A. Khayyer, H. Gotoh, H. Falahaty, Y. Shimizu, Comput. Phys. Comm. 232 (2018) 139–164.
[27] A. Colagrossi, A. Souto-Iglesias, M. Antuono, S. Marrone, Phys. Rev. E 87 (2) (2013) 023302.
[28] A.J.C. Crespo, M. Gómez-Gesteira, R.A. Dalrymple, J. Hydraul. Res. 45 (5) (2007) 631–642.
[29] F. Zhang, A. Crespo, C. Altomare, J. Domínguez, A. Marzeddu, S. Shang, M. Gómez-Gesteira, J. Hydrodyn. 30 (1) (2018) 95–105, http://dx.doi.org/10.1007/s42241-018-0010-0.
[30] A.J.C. Crespo, C. Altomare, J.M. Domínguez, J. González-Cao, M. Gómez-Gesteira, Coast. Eng. (126) (2017) 11–26.
[31] J.C. Marongiu, F. Leboeuf, J. Caro, E. Parkinson, J. Hydraul. Res. 47 (2010) 40–49.
[32] D.J. Price, J. Comput. Phys. 231 (3) (2012) 759–794.
[33] J.L. Cercos-Pita, G. Bulian, L. Pérez-Rojas, A. Francescutto, Ocean Eng. 120 (2016) 281–288, http://dx.doi.org/10.1016/j.oceaneng.2016.03.015.
[34] J.L. Cercos-Pita, M. Antuono, A. Colagrossi, A. Souto-Iglesias, Comput. Methods Appl. Mech. Engrg. 317 (2017) 771–791, http://dx.doi.org/10.1016/j.cma.2016.12.037.
[35] J.M. Dominguez, A.J.C. Crespo, M. Hall, C. Altomare, M. Wu, V. Stratigaki, P. Troch, L. Cappietti, M. Gomez-Gesteira, Coast. Eng. 153 (2019) http://dx.doi.org/10.1016/j.coastaleng.2019.103560.
[36] A. Colagrossi, M. Landrini, J. Comput. Phys. 191 (2) (2003) 448–475.
[37] X.Y. Hu, N.A. Adams, J. Comput. Phys. 213 (2) (2006) 844–861, http://dx.doi.org/10.1016/j.jcp.2005.09.001.
[38] T. Fonty, M. Ferrand, A. Leroy, A. Joly, D. Violeau, Int. J. Multiph. Flow. 111 (2019) 158–174, http://dx.doi.org/10.1016/j.ijmultiphaseflow.2018.11.007.
[39] A.M. Abdelrazek, I. Kimura, Y. Shimizu, J. Glaciol. 62 (232) (2016) 335–347.
[40] H. Bui Ha, R. Fukagawa, K. Sako, S. Ohno, Int. J. Numer. Anal. Methods Geomech. 32 (2008) 1537–1570.

[41] A.J.C. Crespo, J.M. Domínguez, B.D. Rogers, M. Gómez-Gesteira, S. Longshaw, R. Canelas, R. Vacondio, A. Barreiro, O. García-Feal, Comput. Phys. Comm. 187 (2015) 204–216.
[42] J.P. Vila, Math. Models Methods Appl. Sci. 9 (2) (1999) 161–209.
[43] L.M. Armstrong, S. Gu, K.H. Luo, Int. J. Heat Mass Transfer 53 (21–22) (2010) 4949–4959.
[44] V. Kumaran, C. R. Phys. 16 (2015) 51–61.
[45] D.G. Schaeffer, J. Differential Equations 66 (1987) 19–50.
[46] K. Terzaghi, Theoretical Soil Mechanics, Wiley, New York, London, 1943.
[47] L.C. Van Rijn, Principles of Sediment Transport in Rivers, Estuaries, and Coastal Seas, Aqua Publications, 1993.
[48] D. Violeau, A. Leroy, J. Comput. Phys. 256 (2014) 388–415.
[49] Paraview (Kitware), https://github.com/Kitware/ParaView.
[50] OpenFOAM (OpenCFD Ltd), 2019, https://github.com/OpenFOAM/OpenFOAM-dev. (Last Accessed 28 May 2019), https://github.com/isoAdvector/isoAdvector. (Last Accessed 28 May 2019).
[51] Gnuplot (Williams & Kelley), http://www.gnuplot.info/.
[52] GSView (Ghostgum Software Pty Ltd), https://www.ghostscript.com/.
[53] Engauge Digitizer (Mitchell et al.), https://github.com/markummitchell/engauge-digitizer.
[54] Image Magick (ImageMagick Studio LLC), 2019, https://www.imagemagick.org, (Last Accessed 28 May 2019).
[55] Virtual Dub (Avery Lee), 2019, http://www.virtualdub.org/, (Last Accessed 28 May 2019).
[56] SRTM3/DTED1 (USGS), 2019, http://earthexplorer.usgs.gov/, (Last Accessed 28 May 2019).
[57] M. Rexer, C. Hirt, Aust. J. Earth Sci. 61 (2) (2014) 213–226, http://dx.doi.org/10.1080/08120099.2014.884983.
[58] GDAL (OSGEO), https://github.com/OSGeo/gdal.
[59] B. Zoppé, Simulation numérique et analyse de l'écoulement dans les auguets des turbines Pelton (Ph.D. thesis), Institut national polytechnique, Grenoble, France, 2004.
[60] R. Albano, A. Sole, D. Mirauda, J. Adamowski, J. Hydrol. 541 (A) (2016) 344–358.
[61] S. Manenti, A. Amicarelli, S. Todeschini, Water 10 (4) (2018) 515, http://dx.doi.org/10.3390/w10040515.
[62] M. Paggi, A. Amicarelli, P. Lenarda, Lubricants 7 (103) (2019) http://dx.doi.org/10.3390/lubricants7120103, Scopus indexing.
[63] A. Hay, A. Leroyer, M. Visonneau, J. Mar. Sci. Technol. 11 (2006) 1–18.
[64] L. Lobovsky, E. Botia-Vera, F. Castellana, J. Mas-Soler, A. Souto-Inglesia, J. Fluid Struct. 48 (2014) 407–434.
[65] Amicarelli A., S. Manenti, M. Paggi, SPH modelling for 3D urban dam-break floods, with full-scale topography and flooding damage model for electrical substations, Int. J. Comput. Fluid Dyn. (submitted).
[66] D. Berzi, F.C. Bossi, E. Larcan, Phys. Rev. E 85 (2012) 051308, 1-5.
[67] G. Viccione, B. Tagliafierro, Simulating dry granular flow with dualsphysics, in: Aronne Armanini, Elena Nucci (Eds.), Proc. of the 5th IAHR Europe Congress - New Challenges in Hydraulic Research and Engineering, Trento, 13–15 June 2018, Italy, pp. 401–402, 2018, http://dx.doi.org/10.3850/978-981-11-2731-1_316-cd.
[68] L. Sarno, L. Carleo, M.N. Papa, P. Villani, Rock Mech. Rock Eng. 51 (1) (2018) 203–225, http://dx.doi.org/10.1007/s00603-017-1311-2.
[69] Rendina I., G. Viccione, L. Cascini, Theor. Comput. Fluid Dyn. 33 (2) (2019) 107–123, https://doi.org/10.1007/s00162-019-00486-y.
[70] D. Arosio, L. Longoni, M. Papini, G. Bievre, L. Zanzi, Landslides 16 (2019) 1257–1271, http://dx.doi.org/10.1007/s10346-019-01176-w.
[71] D. Brambilla, V.I. Ivanov, L. Longoni, D. Arosio, M. Papini, Workshop on World Landslide Forum, Springer, Cham, 2017, pp. 431–437.
[72] A. Almqvist, J. Fabricius, R. Larsson, P. Wall, J. Tribol. 136 (1) (2014) 011706.
[73] M. Paggi, Q.-C. He, Wear 336–337 (2015) 86–95.
[74] M. Paggi, M. Ciavarella, Wear 268 (7–8) (2010) 1020–1029.
[75] SPHERIC (SPH scientific and industrial community affiliated to ERCOFTAC -European Research Community On Flow, Turbulence and Combustion-), 2019, http://spheric-sph.org/. (Last Accessed 28 May 2019), http://spheric-sph.org/sph-projects-and-codes. (Last Accessed 28 May 2019).