SPHERA v.8.0 (RSE SpA): documentation

This documentation file is intended to provide only additional and updated material, beyond the other SPHERA GitHub repository files and the associated papers on International Journals (Sec.1).

1. DESCRIPTION AND REFERENCES

SPHERA v.8.0 (RSE SpA) is free research software (FOSS) based on the SPH ("Smoothed Particle Hydrodynamics") method, which represents a mesh-less Computational Fluid Dynamics technique for free surface and multi-phase flows.

So far, SPHERA has been applied to represent several types of floods (and landslides) with transport of solid bodies and bed-load transport, and sloshing tanks.

With Copyright 2005-2015 (RSE SpA - formerly ERSE SpA, formerly CESI RICERCA, formerly CESI-Ricerca di Sistema -), SPHERA has been developed for RSE SpA (hereafter RSE, unique owner of the patrimonial rights of SPHERA) by the following authors (SPHERA author list): Andrea Amicarelli, Antonio Di Monaco, Sauro Manenti, Elia Giuseppe Bon, Daria Gatti, Giordano Agate, Stefano Falappi, Barbara Flamini, Roberto Guandalini, David Zuccalà.

The main numerical developments featuring SPHERA (so far) are listed in chronological reverse order:

- 3D SPH numerical scheme for the transport of solid bodies in free surface flows. Reference: Amicarelli et al. (2015, CAF, [6]):
  > 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
- 3D SPH numerical scheme for a boundary treatment based on discrete surface and volume elements, and on a 1D Linearized Partial Riemann Solver coupled with a MUSCL (Monotonic Upstream-Centered Scheme for Conservation Laws) spatial reconstruction scheme. Reference: Amicarelli et al. (2013, IJNME, [5]):
  > 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.
- SPH numerical scheme for a 2D erosion criterion. Reference: Manenti et al. (2012, JHE, [98]):
  > 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.
- 3D SPH numerical scheme for a boundary treatment based on volume integrals, which are numerically computed outside of the fluid domain (semi-analytic approach). Reference: Di Monaco et al. (2011, EACFM, [39]):
  > Di Monaco A., Manenti S., Gallati M., Sibilla S., Agate G., Guandalini R., 2011; SPH modeling of solid boundaries through a semi-analytic approach; Engineering Applications of Computational Fluid Mechanics, 5, 1, 1–15.

Other major numerical developments are available in SPHERA (i.e. 3D erosion criterion also with mixture-fixed bed interactions; bed-load transport), but they are preliminary at this stage.

Since its SPHERA v.7.0 branches SPHERA has being developed under a Git repository (GitHub web site). Its current version contains the folders of Table 1.1.

SPHERA is free software released under the GNU General Public License (Free Software Foundation).

The email address to contact the first author of SPHERA is: andrea.amicarelli@rse-web.it .

Folder	Description
(main folder)	License file (GNU-GPL license). Documents on SPHERA registration at SIAE.
doc	Present documentation file.
src	SPHERA source code (with makefile)
bin	SPHERA executable files compiled with gfortran/ifort for run/debug executions
input	Input files for validated test cases (Sec.12). A template for the main input file with comments.

Table 1.1 Folders in SPHERA Git repository.

2. WARRANTIES AND RESPONSABILITIES

SPHERA v.8.0 is released “as is” with no warranty. NEITHER RSE SPA, NOR ANY OF ITS REPRESENTATIVES (OR ANY CODE AUTHOR) MAKE ANY WARRANTY, EXPRESS OR IMPLIED, OR ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS, EFFECTIVENESS, INTEGRITY, AVAILABILITY, OR USEFULNESS OF THE SOFTWARE, ANY INFORMATION PERTAINING TO THE SOFTWARE, OR REPRESENTS THAT ITS USE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS. No support service (for the code installation, use, teaching activities, …) is implied by or included in the software license. ”

3. CITATION OF SPHERA V.8.0

All the published and unpublished items/products/documents of every kind (i.e. results, publications, software, projects, web pages, press and digital documents, teaching or technological devices, reports, dissemination tools/devices,…) related to SPHERA v.8.0 need the following citation: "SPHERA v.8.0 (RSE SpA)".

Further proper citations may refer to SPHERA-related papers on International Peer-Reviewed Journals indexed by Scopus and Web of Science (Sec.1).

It is also mandatory to cite the use of SPHERA in all the related publications, reports and dissemination tools and media (also included press and digital products), by means of the following citation:

"SPHERA v.8.0 is realised by RSE SpA thanks to the funding "Fondo di Ricerca per il Sistema Elettrico" within the frame of a Program Agreement between RSE SpA and the Italian Ministry of Economic Development (Ministero dello Sviluppo Economico)."

4. SPHERA DEVELOPERS/AUTHORS

This section reports few and non-exhaustive notes, which may help potential authors of SPHERA or its derived codes.

If one receives a code with the GNU-GPL license, then she/he has to transmit the license rights unchanged. In particular, it can be useful to remind that GNU-GPL is viral. This also implies that a code, which contains just very few lines of a GNU-GPL code, becomes necessarily a GNU-GPL code in its entirety, when integrating those lines of a GNU-GPL code.

Every modifications of a code derived from a GNU-GPL code must underline every modifications with respect to the original GNU-GPL code.

5. SPHERA OFFICIAL USERS

The information reported in this section only has an educational aim and does not modify the terms and conditions of SPHERA license.

SPHERA v.8.0 is available on GitHub ([151]). Potential developers or users may:

1) contribute to the development of SPHERA as code authors (by means of a free GitHub account; basic knowledge of Git is mandatory);
2) contribute to the validation of SPHERA as “official users” (by means of a free GitHub account; basic knowledge of Git is not mandatory);
3) use SPHERA independently (respecting the code license and citation terms);
4) independently introduce relevant modifications in SPHERA, thus obtaining a FOSS derived code (which has a different name from SPHERA and has to cite SPHERA as the original code) and redistribute it (bound to the GNU-GPL license and in the respect of SPHERA citation terms) or propose it to RSE for its integration in SPHERA;
5) to propose to RSE some program units (not belonging to SPHERA and developed with independent funds), which will be released with GNU-GPL license and integrated in SPHERA;
6) to propose to RSE some program units (of a code developed with independent funds -original code-), which will be released with GNU-LGPL license and integrated in SPHERA, without constraints for the authors to make their original code a FOSS (in its entirety).

The modifications of the source code and the new input files produced with independent funds by non-RSE authors could be proposed to RSE (with a non-RSE Copyright) to be integrated in SPHERA as FOSS program units and input files. In case of acceptance, these contributions will be kept updated by RSE in the following code versions, until RSE will consider them useful for SPHERA development and validation.

SPHERA authors and “official users” need to activate a free account on github.com and “fork” SPHERA, by clicking on the icon “fork” in the official SPHERA public repository ([151; Figure 5.1, Figure 5.2, Figure 5.3, Figure 5.4, Figure 5.5, Figure 5.6).

The basic knowledge of Git is mandatory only for SPHERA authors. In this context, the following links may be useful:

- https://git-scm.com/
- https://www.youtube.com/watch?v=U8GBXvdmHT4&index=3&list=PLg7s6cbtAD15Das5LK9mXt_g59DLWxKUe

Anyone could be informed on the real-time code upgrades by means of automatic emails sent by GitHub. This free service is available by activating a free account at GitHub (https://github.com/join) and then clicking on the icon “watch” in the official public repository of SPHERA. When activating a GitHub account, it could be convenient to choose a user name, which included name, surname and affiliation. This will permit to get recognized and attend to SPHERA development/validation (the symbol “.” is not permitted within a GitHub user name).

Finally, SPHERA is indexed in the list of SPH codes of SPHERIC (SPH scientific Community; Figure 5.7).

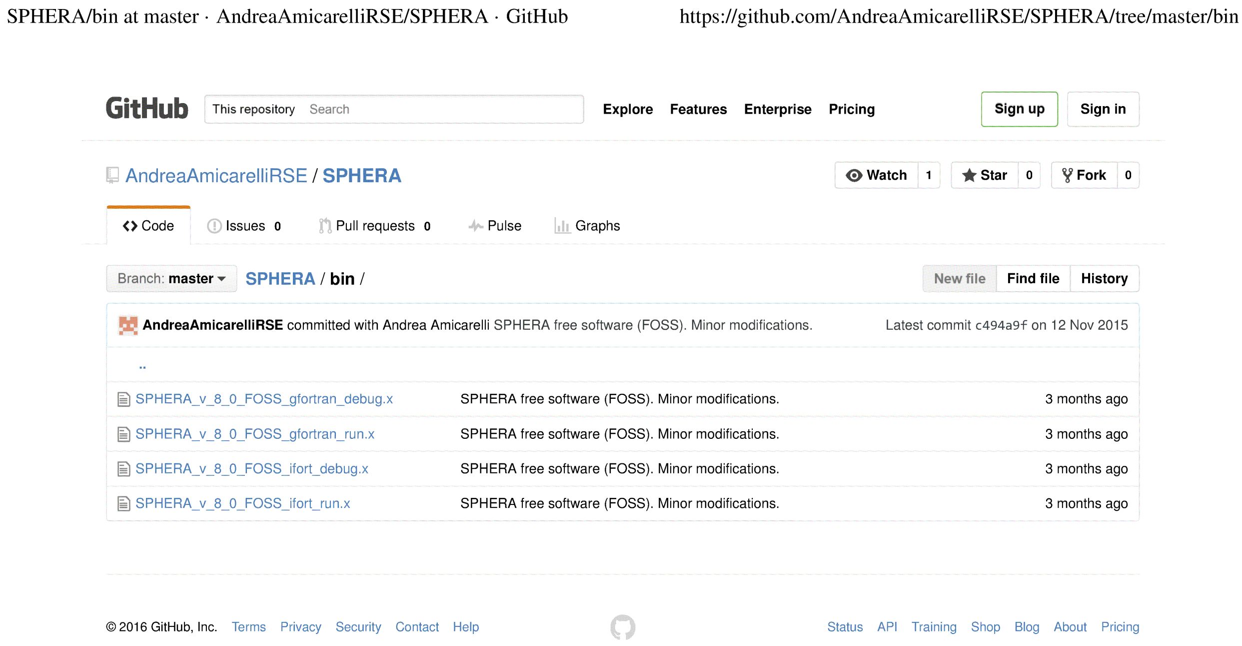


Figure 5.1. SPHERA on GitHub: master branch. Executable codes.

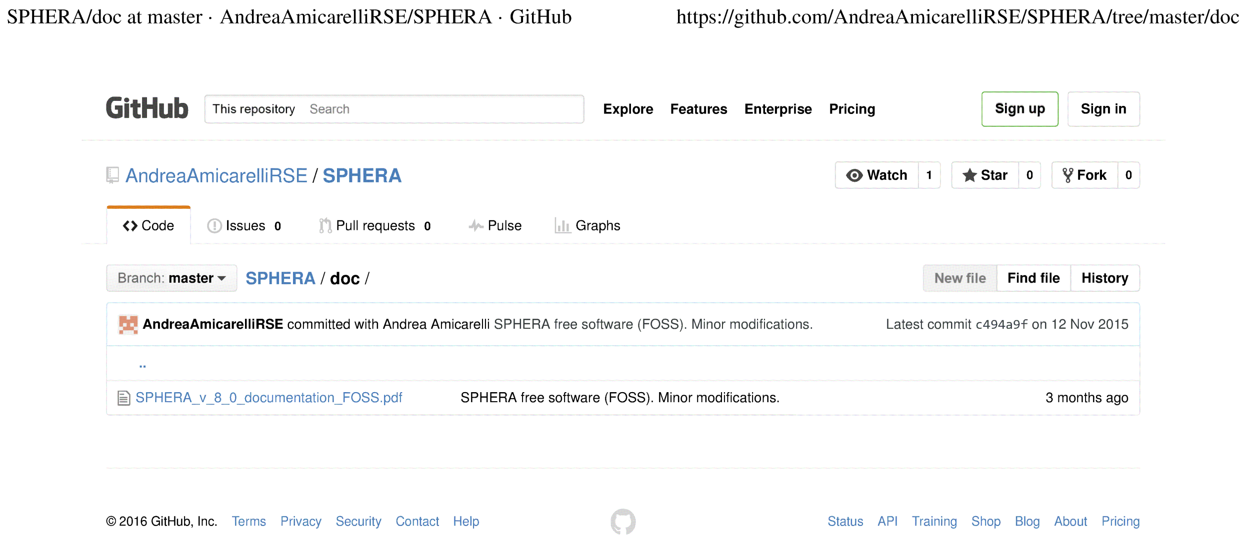


Figure 5.2. SPHERA on GitHub: master branch. Documentation.

Releases · AndreaAmicarelliRSE/SPHERA · GitHub https://github.com/AndreaAmicarelliRSE/SPHERA/releases

GitHub This repository Search Explore Features Enterprise Pricing Sign up Sign in

AndreaAmicarelliRSE / **SPHERA** Watch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Pulse Graphs

Releases Tags

Latest release

v.8.0

55981ad

SPHERA v.8.0 (RSE SpA)

AndreaAmicarelliRSE released this on 19 Nov 2015

`SPHERA free software (FOSS). Signed SIAE registration application.`

Downloads

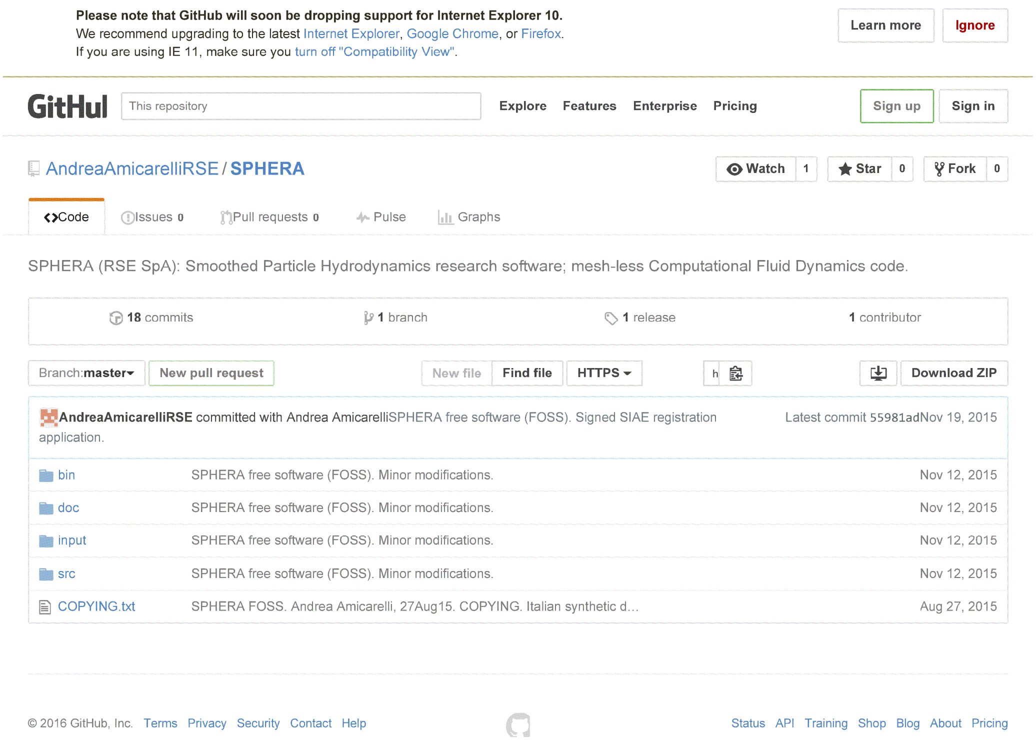
Source code (zip)

Source code (tar.gz)

© 2016 GitHub, Inc. Terms Privacy Security Contact Help Status API Training Shop Blog About Pricing

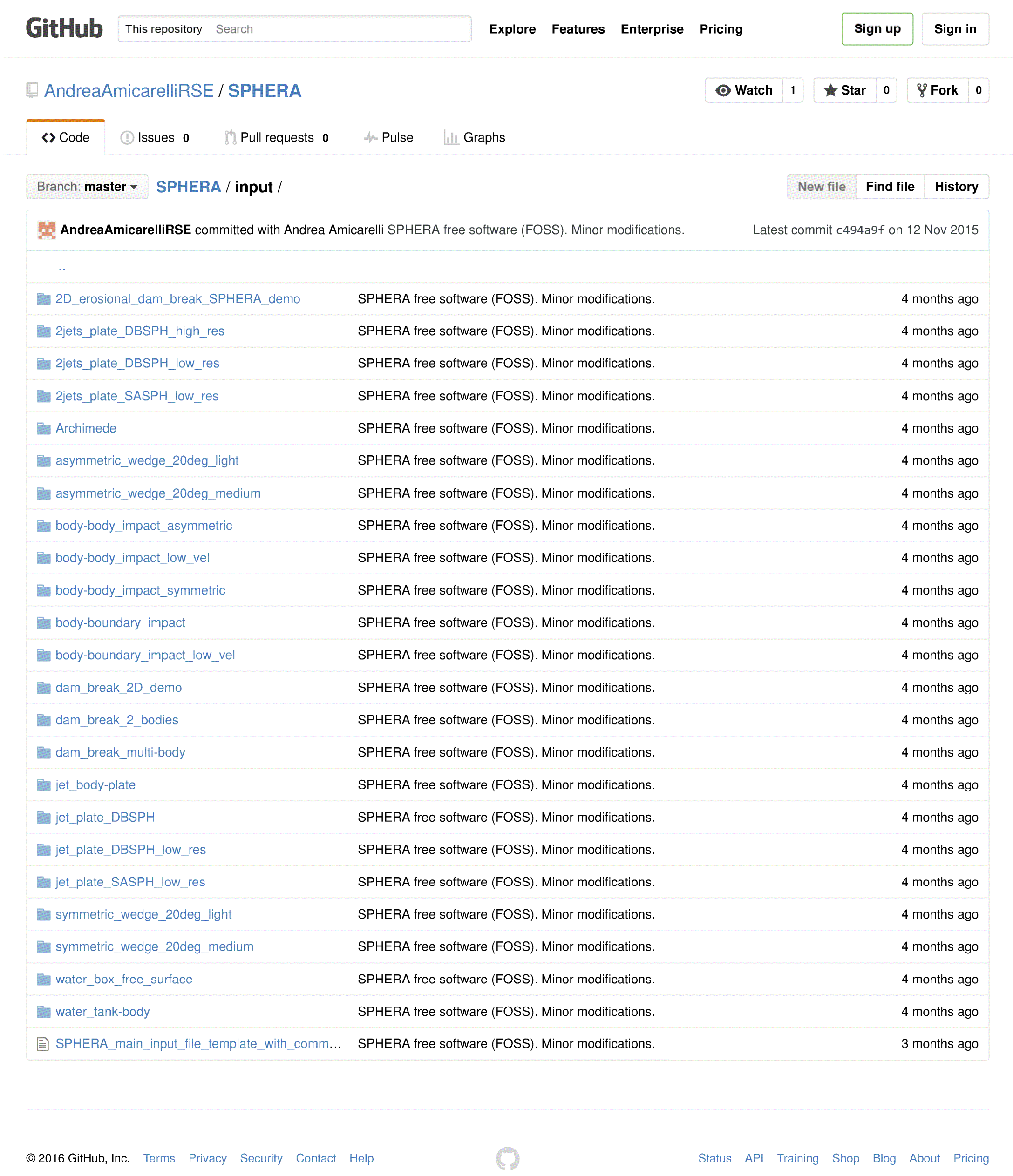
1 di 1 22/02/2016 15:04

Figure 5.3. SPHERA on GitHub. First FOSS release (SPHERA v.8.0).



https://github.com/AndreaAmicarelliRSE/SPHERA 22/02/2016

Figure 5.4. SPHERA on GitHub: master branch or trunk.


GitHub This repository Search Explore Features Enterprise Pricing Sign up Sign in

AndreaAmicarelliRSE / SPHERA Watch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Pulse Graphs

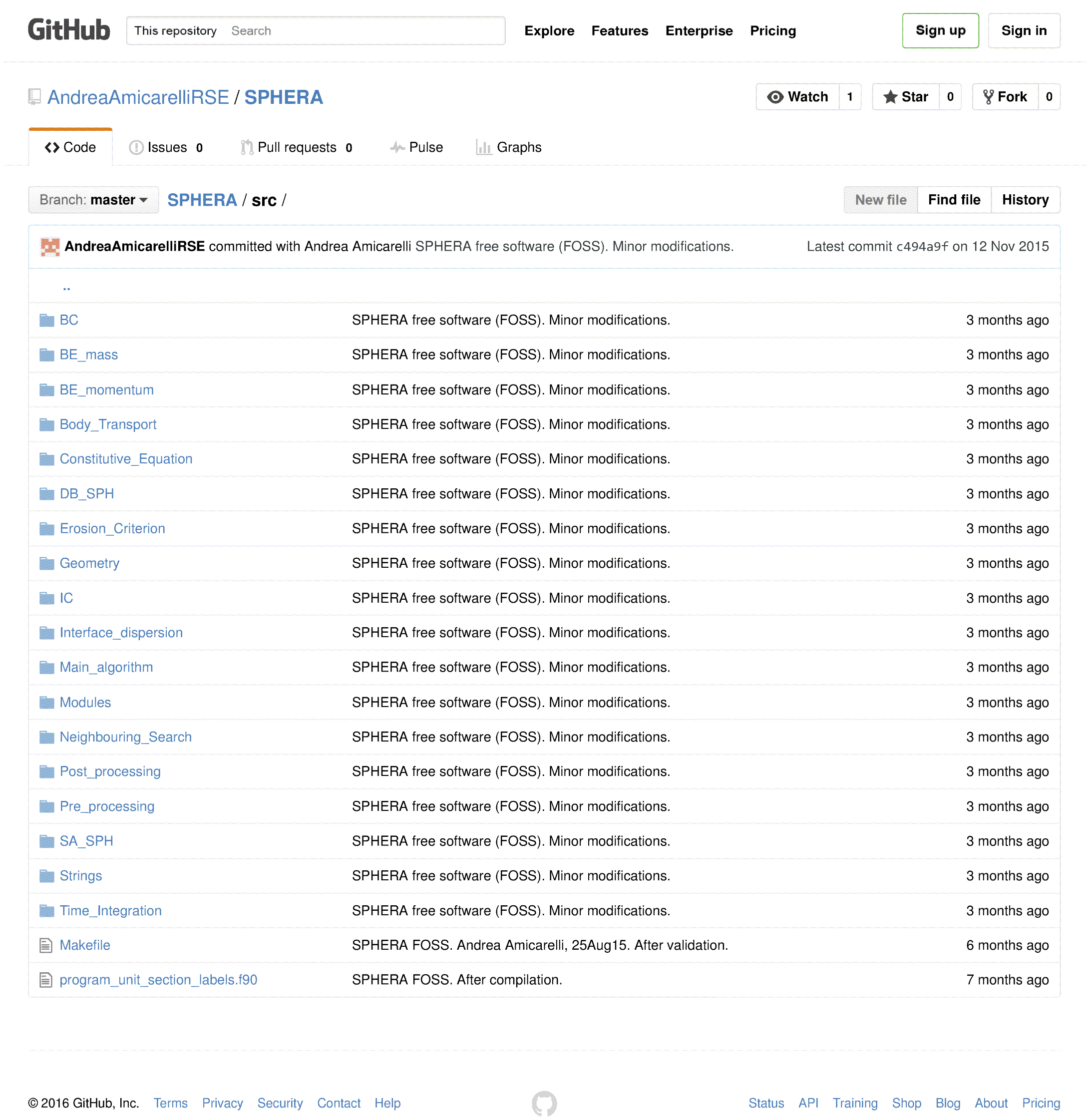
Branch: master SPHERA / input / New file Find file History

AndreaAmicarelliRSE committed with Andrea Amicarelli SPHERA free software (FOSS). Minor modifications. Latest commit c494a9f on 12 Nov 2015

..		
2D_erosional_dam_break_SPHERA_demo	SPHERA free software (FOSS). Minor modifications.	4 months ago
2jets_plate_DBSPH_high_res	SPHERA free software (FOSS). Minor modifications.	4 months ago
2jets_plate_DBSPH_low_res	SPHERA free software (FOSS). Minor modifications.	4 months ago
2jets_plate_SASPH_low_res	SPHERA free software (FOSS). Minor modifications.	4 months ago
Archimede	SPHERA free software (FOSS). Minor modifications.	4 months ago
asymmetric_wedge_20deg_light	SPHERA free software (FOSS). Minor modifications.	4 months ago
asymmetric_wedge_20deg_medium	SPHERA free software (FOSS). Minor modifications.	4 months ago
body-body_impact_asymmetric	SPHERA free software (FOSS). Minor modifications.	4 months ago
body-body_impact_low_vel	SPHERA free software (FOSS). Minor modifications.	4 months ago
body-body_impact_symmetric	SPHERA free software (FOSS). Minor modifications.	4 months ago
body-boundary_impact	SPHERA free software (FOSS). Minor modifications.	4 months ago
body-boundary_impact_low_vel	SPHERA free software (FOSS). Minor modifications.	4 months ago
dam_break_2D_demo	SPHERA free software (FOSS). Minor modifications.	4 months ago
dam_break_2_bodies	SPHERA free software (FOSS). Minor modifications.	4 months ago
dam_break_multi-body	SPHERA free software (FOSS). Minor modifications.	4 months ago
jet_body-plate	SPHERA free software (FOSS). Minor modifications.	4 months ago
jet_plate_DBSPH	SPHERA free software (FOSS). Minor modifications.	4 months ago
jet_plate_DBSPH_low_res	SPHERA free software (FOSS). Minor modifications.	4 months ago
jet_plate_SASPH_low_res	SPHERA free software (FOSS). Minor modifications.	4 months ago
symmetric_wedge_20deg_light	SPHERA free software (FOSS). Minor modifications.	4 months ago
symmetric_wedge_20deg_medium	SPHERA free software (FOSS). Minor modifications.	4 months ago
water_box_free_surface	SPHERA free software (FOSS). Minor modifications.	4 months ago
water_tank-body	SPHERA free software (FOSS). Minor modifications.	4 months ago
SPHERA_main_input_file_template_with_comm…	SPHERA free software (FOSS). Minor modifications.	3 months ago

© 2016 GitHub, Inc. Terms Privacy Security Contact Help Status API Training Shop Blog About Pricing

Figure 5.5. SPHERA on GitHub: master branch. Input files.


GitHub This repository Search Explore Features Enterprise Pricing Sign up Sign in

AndreaAmicarelliRSE / SPHERA Watch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Pulse Graphs

Branch: master SPHERA / src / New file Find file History

AndreaAmicarelliRSE committed with Andrea Amicarelli SPHERA free software (FOSS). Minor modifications. Latest commit c494a9f on 12 Nov 2015

..		
BC	SPHERA free software (FOSS). Minor modifications.	3 months ago
BE_mass	SPHERA free software (FOSS). Minor modifications.	3 months ago
BE_momentum	SPHERA free software (FOSS). Minor modifications.	3 months ago
Body_Transport	SPHERA free software (FOSS). Minor modifications.	3 months ago
Constitutive_Equation	SPHERA free software (FOSS). Minor modifications.	3 months ago
DB_SPH	SPHERA free software (FOSS). Minor modifications.	3 months ago
Erosion_Criterion	SPHERA free software (FOSS). Minor modifications.	3 months ago
Geometry	SPHERA free software (FOSS). Minor modifications.	3 months ago
IC	SPHERA free software (FOSS). Minor modifications.	3 months ago
Interface_dispersion	SPHERA free software (FOSS). Minor modifications.	3 months ago
Main_algorithm	SPHERA free software (FOSS). Minor modifications.	3 months ago
Modules	SPHERA free software (FOSS). Minor modifications.	3 months ago
Neighbouring_Search	SPHERA free software (FOSS). Minor modifications.	3 months ago
Post_processing	SPHERA free software (FOSS). Minor modifications.	3 months ago
Pre_processing	SPHERA free software (FOSS). Minor modifications.	3 months ago
SA_SPH	SPHERA free software (FOSS). Minor modifications.	3 months ago
Strings	SPHERA free software (FOSS). Minor modifications.	3 months ago
Time_Integration	SPHERA free software (FOSS). Minor modifications.	3 months ago
Makefile	SPHERA FOSS. Andrea Amicarelli, 25Aug15. After validation.	6 months ago
program_unit_section_labels.f90	SPHERA FOSS. After compilation.	7 months ago

© 2016 GitHub, Inc. Terms Privacy Security Contact Help Status API Training Shop Blog About Pricing

Figure 5.6. SPHERA on GitHub: master branch. Source code.

Florian Fleissner (Inpartik, florian.fleissner@inpartik.com). For further information on Pasimodo and some example videos please visit http://www.itm.uni-stuttgart.de/research/pasimodo/pasimodo_en.php (http://www.itm.uni-stuttgart.de/research/pasimodo/pasimodo_en.php).

SPH-flow

SPH-flow (http://www.sph-flow.com/) is an academic-industrial consortium innovation in multiphysics and fluid SPH simulations.

SPHERA

SPHERA v.8.0 (https://github.com/AndreaAmicarelliRSE/SPHERA)(RSE SpA) is SPH research and free software. So far (2015), SPHERA has been characterized by two alternative boundary treatment schemes (based on either volume integrals or discrete surface elements), a 2D erosion criterion, a scheme for body transport in free surface flows. SPHERA has represented several types of floods (with transport of solid bodies and granular material) and sloshing tanks. SPHERA is published and developed as FOSS (Free/Libre and Open Source Software) on GitHub at https://github.com/AndreaAmicarelliRSE/SPHERA (https://github.com/AndreaAmicarelliRSE/SPHERA).

SPHysics

SPHYSICS (http://wiki.manchester.ac.uk/sphysics) - This is a free open-source SPH code that released 2007 developed jointly by researchers at the Johns Hopkins University (U.S.A.), the University of Vigo (Spain), the University of Manchester (U.K.) and the University of Rome La Sapienza (Italy). The 2-D & 3-D code has been developed specifically for free-surface hydrodynamics. The code now includes **serial, parallel and GPU versions**.

SPLASH

SPLASH is a publicly available visualisation tool for Smoothed Particle Hydrodynamics simulations developed over a number of years, and can be used to read, convert and visualise output from all known publicly available SPH codes. Website: http://users.monash.edu.au/~dprice/splash (http://users.monash.edu.au/~dprice/splash).

About

SPHERIC is the international organisation representing the community of researchers and industrial users of Smoothed Particle Hydrodynamics (SPH).

spheric

Figure 5.7. SPHERA indexed at SPHERIC web site (February 2016, extract from the list of the 13 SPH codes indexed at http://spheric-sph.org/sph-projects-and-codes).

6. THE NUMERICAL MODEL: TRANSPORT OF SOLID BODIES IN FREE SURFACE FLOWS AND SEMI-ANALYTIC APPROACH

Aftyer a brief introduction to Smoothed Particle Hydrodynamics (SPH), this section describes the balance equations for fluid (Sec.6.2) and body (Sec.6.3) dynamics and the 2-way interaction terms related to both fluid-body (Sec. 6.4) and solid-solid (Sec.6.5) interactions.
Please refer to SPHERA main references (Sec.1) for further details.

6.1. Smoothed Particle Hydrodynamics (SPH)

Smoothed Particle Hydrodynamics (SPH) represents a mesh-less CFD technique, 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:

$$\langle f \rangle_{I,\underline{x}_0} \equiv \int_{V_h} fWdx^3 \tag{6.1}$$

where W is the kernel function ([7]), $\underline{x}_0$ the position of a generic computational point and V_h the integration volume, which is called kernel support. This is represented by a sphere of radius 2h, 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 (6.1), after replacing f with the targeted derivative. After integration by parts, one obtains:

$$\left\langle \left. \frac{\partial f}{\partial x_i} \right| \right\rangle_{I,\underline{x}_0} = \int_{A_h} fWn_i dx^2 - \int_{V_h} f \frac{\partial W}{\partial x_i} dx^3 \tag{6.2}$$

The integration also involves the surface A_h of the kernel support. The associated surface integral is non-zero in case of a truncated kernel support. The representation of this term noticeably differentiates SPH codes (Adami et al., 2012, [1]; Hashemi et al., 2012, [64]; Macia et al., 2012, [97]; Mayrhofer et al., 2013, [105]; Ferrand et al., 2013, [49]; Amicarelli et al., 2013, [5]).
Far from boundaries, the SPH particle approximation of (6.2) reads:

$$\left\langle \left. \frac{\partial f}{\partial x_i} \right| \right\rangle_{\underline{x}_0} = -\sum_b f_b \left. \frac{\partial W}{\partial x_i} \right|_b \omega_b \tag{6.3}$$

where a summation on particle volumes (ω) 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.
Usually, the approximation (6.3) is replaced by more complicated and accurate formulas. Further, the SPH technique can also approximate a generic n-th derivative, following the same approach of the cited equation.
SPH approximations can discretize the functions and derivatives in the fluid dynamics balance equations by means of a particle Lagrangian mesh-less technique.
SPH technique is characterized by several advantages: a direct estimation of the position of free surface and multi-phase interfaces; an effective representation of body dynamics transported in fluid flows; a direct estimation of Lagrangian derivatives (absence of non-linear terms on the Left-Hand Side of the balance equations); reliable simulations of fast transient phenomena; absence of a computational mesh (mesh-less); simple algorithms (for Weakly Compressible - SPH codes, possible convergence algorithms only refer to specific schemes and are little time consuming).

The main applications of SPH models refer to the following topics: floods (Vacondio et al., 2012, [168]; Amicarelli et al., 2015, [6]; Di Monaco et al., 2011, [39]); sloshing tanks (Souto-Iglesias et al., 2006, [155]; Delorme et al., 2009, [35]; Amicarelli et al., 2013, [5]); wave motion (e.g., Patel et al., 2009, [127]; Antuono et al., 2011, [11]; Liu et al., 2013, [94]; Omidvar et al., 2012a, [122]); hydraulic turbines (Marongiu et al., 2010, [100]); fast landslides (e.g., Kumar et al., 2013, [79]); erosion and bed-load transport (Manenti et al., 2012, [98]); liquid jets (e.g., Koukouvinis et al., 2013, [77]); pollutant dispersion; astrophysics (e.g., Price & Monaghan, 2007, [134]); magneto-fluid dynamics (e.g., Price, 2012, [132]); multi-phase and multi-fluid flows (e.g., Kajtar & Monaghan, 2012, [72]).

6.2. SPH approximation of the balance equations of fluid dynamics with a boundary treatment based on the semi-analytic approach of Di Monaco et al. (2011)

This section relies on [39] and [6], which are suggested for further details.
The numerical scheme for the main flow is a Weakly-Compressible (WC) SPH model, which takes benefit from a boundary treatment based on the semi-analytic approach of Vila (1999, [174]). Its basic features are deeply described in Di Monaco et al. (2011, [39]) and here briefly reported.
Consider Euler's momentum and continuity equations, in the following forms:

$$\frac{du_i}{dt} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} - \delta_{i3}g = -\frac{\partial\left(p/\rho\right)}{\partial x_i} - \frac{p}{\rho^2}\frac{\partial \rho}{\partial x_i} - \delta_{i3}g, \qquad i = 1,2,3$$

$$\frac{d\rho}{dt} = -\rho\nabla\cdot\underline{u} \tag{6.4}$$

where $\underline{u} \equiv (u,v,w)$ is the velocity vector, p pressure, ρ fluid density, δ_{ij} Kronecker's delta, $\underline{x}$ position and t time. We need to compute (6.4) at each fluid particle position by using the SPH formalism and taking into account the boundary terms (fluid-frontier and fluid-body interactions), as described in the following.
Consider the discretization of (6.4), as provided by the SPH approximation of the first derivative of a generic function (f), according to the semi-analytic approach ("$_{SA}$"; [174]):

$$\left\langle \frac{\partial f}{\partial x_i} \right\rangle_{SA,0} = \sum_b (f_b - f_0)\frac{\partial W_b}{\partial x_i}\omega_b + \int_{V_h'} (f - f_0)\frac{\partial W}{\partial x_i}dx^3 \tag{6.5}$$

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 (6.5) 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 (6.5) represents the boundary term, which is a convolution integral on the truncated portion of the kernel support. In this fictitious and outer volume (V_h'), one needs to define the generic function f (pressure, velocity or density alternatively).
The semi-analytic approach ("$_{SA}$") (the version of [39]) hypothesizes the following linearization and assumptions to compute f in V_h':

$$f \cong f_{SA} + \left.\frac{\partial f}{\partial x_i}\right|_{SA}(\underline{x} - \underline{x}_0) \Rightarrow \left\langle \frac{\partial f}{\partial x_i} \right\rangle_{SA} = \sum_b (f_b - f_0)\frac{\partial W_b}{\partial x_i}\omega_b + \int_{V_h'} f_{SA}\frac{\partial W}{\partial x_i}dx^3 + \int_{V_h'} \left.\frac{\partial f}{\partial x_i}\right|_{SA}(\underline{x} - \underline{x}_0)\frac{\partial W}{\partial x_i}dx^3 \tag{6.6}$$

The peculiar "$_{SA}$" values of the functions and derivatives in V_h' are assigned to represent a null normal gradient of reduced pressure at the frontier interface (while considering uniform density):

$$p_{SA} = p_0, \quad \left\langle \frac{\partial p}{\partial x_i} \right\rangle_{SA} = -\delta_{i3} g; \quad \rho_{SA} = \rho_0, \quad \left\langle \frac{\partial \rho}{\partial x_i} \right\rangle_{SA} = 0 \tag{6.7}$$

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 in the sum of a vector normal to boundary $(\underline{u}_{SA,n})$ and a tangential vector $(\underline{u}_{SA,T})$. 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):

$$\left.\begin{array}{l} \underline{u}_{SA} = \underline{u}_{SA,T} + \underline{u}_{SA,n} \equiv u_{0,T} + [(2\underline{u}_w - \underline{u}_0)\cdot \underline{n}]\underline{n} \\ \underline{u}_{SA,T} \equiv \underline{u}_{0,T}, \quad \left\langle \frac{\partial u_i}{\partial x_i} \right\rangle_{SA} = 0 \end{array}\right\} \Rightarrow \underline{u} - \underline{u}_0 = \underline{u}_{SA} - \underline{u}_0 = 2[(\underline{u}_w - \underline{u}_0)\cdot \underline{n}]\underline{n} \tag{6.8}$$

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 works for "$_j$"), using the semi-analytic approach as a boundary treatment:

$$\left\langle \frac{d\rho}{dt} \right\rangle_0 = \sum_b \rho_b (u_{b,j} - u_{0,j}) \left.\frac{\partial W}{\partial x_j}\right|_b \omega_b + 2\rho_0 \int_{V_h'} [(\underline{u}_w - \underline{u}_0)\cdot \underline{n}] n_j \frac{\partial W}{\partial x_j} dx^3 + C_s \tag{6.9}$$

where $\underline{C}_s$ is introduced to represent a fluid-body interaction term.
On the other hand, we can analogously derive the approximation of the momentum equation (the notation $\langle \ \rangle$ indicates the SPH particle -discrete- approximation):

$$\begin{aligned} \left\langle \frac{du_i}{dt} \right\rangle_0 = & -\delta_{i3} g + \sum_b \left(\frac{p_b}{\rho_b^2} + \frac{p_0}{\rho_0^2} \right) W_b' m_b + 2\frac{p_0}{\rho_0} \int_{V_h'} \frac{\partial W}{\partial x_i} dx^3 + \\ & - \upsilon_M \sum_b \frac{m_b}{\rho_0 r_{0b}^2} (\underline{u}_b - \underline{u}_0)\cdot(\underline{x}_b - \underline{x}_0) \left.\frac{\partial W}{\partial x_i}\right|_b - 2\upsilon_M (\underline{u}_w - \underline{u}_0)\cdot \int_{V_h'} \frac{1}{r_{0w}^2} (\underline{x} - \underline{x}_0) \frac{\partial W}{\partial x_i} dx^3 + \underline{a}_s \end{aligned} \tag{6.10}$$

where $\underline{a}_s$ represents a new acceleration term due to the fluid-body interactions, ν_M is the artificial viscosity (Monaghan, 2005, [111]), m the particle mass and r the relative distance between the neighbouring and the computational particle.
Finally, a barotropic equation of state (EOS) is linearized as follows:

$$p \cong c_{ref}^2 (\rho - \rho_{ref}) \tag{6.11}$$

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

6.3. SPH balance equations for rigid body transport

This section relies on [6], which is suggested for further details.
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:

$$\frac{d\underline{u}_{CM}}{dt} = \frac{\underline{F}_{TOT}}{m_B}$$

$$\frac{d\underline{x}_{CM}}{dt} = \underline{u}_{CM}$$

$$\underline{M}_{TOT} = \underline{\underline{I}}_C \frac{d\underline{\chi}_B}{dt} + \frac{d\underline{\underline{I}}_C}{dt}\underline{\chi}_B = \underline{\underline{I}}_C \frac{d\underline{\chi}_B}{dt} + \underline{\chi}_B \times \left(\underline{\underline{I}}_C \underline{\chi}_B\right) \Rightarrow \frac{d\underline{\chi}_B}{dt} = \underline{\underline{I}}_C^{-1}\left[\underline{M}_{TOT} - \underline{\chi}_B \times \left(\underline{\underline{I}}_C \underline{\chi}_B\right)\right] \tag{6.12}$$

$$\frac{d\underline{\alpha}}{dt} = \underline{\chi}_B$$

Here the subscript "$_B$" refers to a generic computational body and "$_{CM}$" to its centre of mass.
The first two formulas of (6.12) represent the balance equations for the momentum and the time law for the position of the body barycentre ($\underline{F}_{TOT}$ is the global/resultant force acting on the solid). The last two formulas of (6.12) express the balance equation of the angular momentum (χ_B denotes the angular velocity of the generic body) and the time evolution of the solid orientation ($\underline{\alpha}$ is the vector of the angles lying between the body axis and the global reference system). $\underline{M}_{TOT}$ represents the associated torque acting on the body and $\underline{\underline{I}}_C$ the matrix of the moment of inertia of the computational body (Einstein's notation works for the subscript "$_l$"):

$$I_{c,ij} = \int_{V_B} \rho\left(r_l^2 \delta_{ij} - r_i r_j\right) dV = \begin{cases} \int_{V_B} \rho\left(r_k^2 + r_n^2\right) dV, i = j; k, n \neq i \\ -\int_{V_B} \rho\left(r_i r_j\right) dV, i \neq j \end{cases} \tag{6.13}$$

In this sub-section, $\underline{r}$ implicitly represents the relative distance from the body centre of mass.
In order to solve the system (6.12), we need to model the global force and torque, as described in the following. The resultant force is composed of several terms:

$$\underline{F}_{TOT} = \underline{G} + \underline{P}_F + \underline{T}_F + \underline{P}_S + \underline{T}_S, \qquad \underline{T}_F + \underline{T}_S \cong 0 \tag{6.14}$$

$\underline{G}$ represents the gravity force, while $\underline{P}_F$ and $\underline{T}_F$ the vector sums of the pressure and shear forces provided by the fluid. Analogously, $\underline{P}_S$ and $\underline{T}_S$ are the vector sums of the normal and the shear forces provided by other bodies or boundaries (solid-solid interactions). As this study focuses on inertial and quasi-inertial fluid flows, we do not implement neither turbulence scheme nor tangential stresses (simplifying hypothesis). Future works are needed to extend the formulation to a wider category of fluid flows.
The fluid-solid interaction is expressed by the following pressure force:

$$\underline{P}_F = \sum_s p_s A_s \underline{n}_s \tag{6.15}$$

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).

The pressure of a body particle is computed by means of the boundary treatment of Adami et al. (2012, [1]), here implemented and adapted as described in Sec.6.3. Further, the solid-solid interaction term ($\underline{P}_s$) is presented in Sec.6.4.
On the other hand, the torque in (6.12) 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:

$$\underline{M}_{TOT} = \sum_s \underline{r}_s \times \underline{F}_s \tag{6.16}$$

Time integration of the equations in (6.12) is performed using a Leapfrog scheme synchronized with the fluid dynamics balance equations. This means that the body particle pressure is computed simultaneously to fluid pressure, so that this parameter is staggered of around dt/2 with respect to all the other body particle parameters.
After time integration, the model obtains the velocity of a body particle as the sum of the velocity of the corresponding body barycentre and the relative velocity:

$$\underline{u}_s = \underline{u}_{CM} + \underline{\chi}_B \times \underline{r}_s \tag{6.17}$$

Finally, the model updates the body particle normal vectors and absolute positions, according to the following kinematics formulas ($\underline{d\alpha}$ is the increment in the body rotation angle during the on-going time step and R_{ij} the body rotation matrix):

$$\underline{n}_s(t+dt) = \underline{\underline{R}}_B \underline{n}_s(t), \qquad \underline{x}_s(t+dt) = \underline{x}_{CM}(t+dt) + \underline{\underline{R}}_B \underline{r_s}(t)$$

$$\underline{\underline{R}}_B = \underline{\underline{R}}_x \underline{\underline{R}}_y \underline{\underline{R}}_z, \qquad \underline{d\alpha}_B = \underline{\omega}_B dt$$

$$\underline{\underline{R}}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(d\alpha_x) & \sin(d\alpha_x) \\ 0 & -\sin(d\alpha_x) & \cos(d\alpha_x) \end{bmatrix}, \quad \underline{\underline{R}}_y = \begin{bmatrix} \cos(d\alpha_y) & 0 & -\sin(d\alpha_y) \\ 0 & 1 & 0 \\ \sin(d\alpha_y) & 0 & \cos(d\alpha_y) \end{bmatrix}, \quad \underline{\underline{R}}_z = \begin{bmatrix} \cos(d\alpha_z) & \sin(d\alpha_z) & 0 \\ -\sin(d\alpha_z) & \cos(d\alpha_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.18}$$

6.4. Fluid-body interaction terms

This section relies on [6], which is suggested for further details.
The fluid-body interaction terms rely on the boundary technique introduced by Adami et al. (2012, [1]), here implemented and adapted for free-slip conditions. If boundary is fixed, this method can be interpreted as a discretization of the semi-analytic approach used to treat fluid-boundary interactions (Sec.6.2). The outer domain of (6.5) is here represented by all the body particles inside the kernel support of the computational fluid particle. Further, Adami et al. (2012, [1]) introduce a new term, related to the acceleration of the fluid-solid interface, which influences the estimation of body particle pressure. The implementation and our modifications of this technique is hereafter described.
The fluid-body interaction term in the continuity equation represents a discrete approximation of the analogous term in (6.9), used to treat solid frontiers (free-slip conditions):

$$C_s = 2\rho_0 \sum_s \left[(\underline{u}_s - \underline{u}_0) \cdot \underline{n}_s\right] W_s' \omega_s \tag{6.19}$$

Analogously, the fluid-body interaction term in the momentum equation (6.10) assumes the form:

$$\underline{a}_s = \sum_s \left(\frac{p_s + p_0}{\rho_0^2} \right) W_s' m_s \tag{6.20}$$

The pressure value of the generic neighbouring surface body particle "$_s$" is derived as follows. Consider a generic point at a generic fluid-body interface. In case of free-slip conditions, the normal projection of the acceleration on the fluid side ("$_f$") and on the solid side ("$_w$") are equal (in-built motion in the direction normal to the interface):

$$\left(\frac{d\underline{u}_f}{dt}\right)\cdot\underline{n}_w = \left(-\frac{1}{\rho_f}\underline{\nabla}p_f + \underline{g}\right)\cdot\underline{n}_w = \underline{a}_w\cdot\underline{n}_w \qquad (6.21)$$

The "wall" acceleration at the position of a generic body particle can then be derived by linearizing (6.21). This depends on the particular computational fluid particle "$_0$" we are considering, so that we can refer to the interaction subscript "$_{s,0}$":

$$\underline{\nabla}p_f\cdot\underline{n}_w = \rho_f\left(-\underline{a}_w\cdot\underline{n}_w + \underline{g}\right)\cdot\underline{n}_w \Rightarrow p_{s0} \approx p_0 + \rho_0\left(\underline{g}-\underline{a}_s\right)\cdot\left(\underline{x}_s - \underline{x}_0\right)\cdot\underline{n}_s \qquad (6.22)$$

One may apply a SPH interpolation over all the pressure values estimated according to (6.22) to derive a unique pressure value for a body particle:

$$p_s = \frac{\sum_0 p_{s0}W_{s0}\left(\frac{m_0}{\rho_0}\right)}{\sum_0 W_{s0}\left(\frac{m_0}{\rho_0}\right)} = \frac{\sum_0 \left[p_0 + \rho_0\left(\underline{g}-\underline{a}_s\right)\cdot\underline{r}_{s0}\cdot\underline{n}_s\right]W_{s0}\left(\frac{m_0}{\rho_0}\right)}{\sum_0 W_{s0}\left(\frac{m_0}{\rho_0}\right)} \qquad (6.23)$$

This pressure value is finally used in (6.21). The formulation provided by (6.19), (6.21) and (6.13) differs from Adami et al. (2012, [1]) because of the presence of $\underline{n}_s$ in (6.23), necessary to represent free-slip conditions.
Only a minority of the body particles represents the body surface, but we also need many inner body particles to estimate p_s. Thus, the model defines the normal vectors for the neighbouring body particles lying inside the bodies, as described by the following algorithm.
For any fluid-body particle interaction, each fluid particle searches for the most representative surface body particle to define $\underline{n}_s$ in (6.23) -"$_{s0}$" interaction-. If the on-going body particle "$_s$" belongs to the body surface, then it is immediately considered as representative. Otherwise, the fluid particle "$_0$" isolates its visible neighbouring surface body particles. Visibility is assessed considering the sign of the projection of the inter-particle distance on the body particle normal. The visible neighbour, which is the closest to the joining segment of particles "$_0$" and "$_s$", is then selected. This particle provides the normal "$\underline{n}_s$" for the fluid-solid particle interaction "$_{s0}$" in (6.23).
The assumption (6.21) relies on the fact that all the involved variables are differentiable in time. This means that this equation cannot properly deal with impulses (infinite accelerations). However, the numerical accelerations of our model are always finite and the solid particle accelerations can be easily used in (6.23). Nevertheless, we prefer defining a maximum threshold for $|\underline{a}_s|$, here equal to 10g.

6.5. Modelling the solid-solid interaction terms

This section relies on [6], which is suggested for further details.
The solid-solid interaction term in (6.14) -$\underline{P}_s$- represents body-body and body-boundary (full elastic) impingement forces, whose time and spatial evolution, in the continuum, is theoretically proportional to Dirac's delta. The numerical model needs to discretize $\underline{P}_s$, as explained hereafter.
The "boundary force particle" method of Monaghan (2005, [111]) defines repulsive forces to represent a conservative full elastic impingement between two SPH interacting particles (of any

medium). In particular, the acceleration $\underline{a}_{bfp,jk}$ of particle "$_j$", due to the impingement with particle "$_k$", is aligned with the inter-particle distance $\underline{r}$ and inversely proportional to its absolute value r:

$$\underline{a}_{bfp,jk} = \frac{f_{bfp}}{r_{jk}} \frac{m_k}{m_j + m_k} \underline{r}_{jk} \qquad (6.24)$$

The analytic function f_{bfp} is symmetric with respect to the impact point. The dependence of (6.24) on the particle masses allows conserving both global momentum $\left(m_j \underline{a}_{bfp,jk} = -m_k \underline{a}_{bfp,kj}\right)$ and kinetic energy (one may notice that $\underline{r}_{jk} = -\underline{r}_{jk}$ and $f_{jk} = f_{kj}$). The formulation works for inter-particle high velocity impacts.
This formulation is here implemented and extended to whole solid bodies (not only particle impingements), even at low velocities, as well as body-frontier interactions.
Consider 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:

$$\underline{P}_S = \sum_K \underline{P}_{BK} + \sum_{K*} \underline{P}_{BK*} \qquad (6.25)$$

Adopting 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 "$_K$":

$$\underline{P}_{BK} = -\alpha_I \sum_j \sum_k \frac{2u^2_{\perp,jk}}{r_{per,jk}} \frac{m_j m_k}{m_j + m_k} \Gamma_{jk} \left(1 - \frac{r_{par,jk}}{dx_s}\right) \underline{n}_k \qquad (6.26)$$

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 (6.26) 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 is the size of the body particles). The weighting function Γ is expressed according to Monaghan (2005, [111]) and depends on q= r_{jk}/h:

$$\Gamma_{jk} = \begin{cases} \frac{2}{3}, & 0 \le q < \frac{2}{3} \\ \left(2q - \frac{3}{2}q^2\right), & \frac{2}{3} \le q < 1 \\ \frac{1}{2}(2-q)^2, & 1 \le q < 2 \\ 0, & 2 \le q \end{cases} \qquad (6.27)$$

The present model introduces two modifications for body-body interactions, with respect to the original formulation of the boundary force particles. The first one concerns the impact velocity $u_{\perp,jk}$, which replaces the term "0.1c" in the formulation of Monaghan (2005, [111]) 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 model considers all the inter-particle interactions recorded while the 2 bodies are approaching. The expression for the impact velocity reads:

$$u_{\perp,jk}(t) = \max_{j,k,t^*}\left\{\left|\left(\underline{u}_j - \underline{u}_k\right)\cdot \underline{n}_k\right|\right\}, \qquad t_0 \le t^* \le t \tag{6.28}$$

where t_0 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. In this case, and only during the approaching phase, (6.28) 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, (6.26) introduces the coefficient α_I. This normalizing parameter corrects discretization errors and better preserves the global momentum and kinetic energy of the body-body system during the impingement. If one omitted α_I, (6.26) would drastically under-estimate the impingement forces if the whole mass of the bodies did not lie within the impact zone (of depth 2h). To avoid this shortcoming, a formulation for α_I is presented hereafter. Consider the absolute value of the impingement force P_s as a function of the global parameters of the bodies, instead of the particle values. This second formulation for P_{BK} is denoted as follows:

$$P_{BK}' \equiv \frac{2u^2_{\perp,BK}}{r_{per,BK}}\frac{m_B m_K}{m_B + m_K}\Gamma_{BK}, \qquad r_{per,BK} = \min_{B,K}\left\{r_{per,jk}\right\}, \qquad u^2_{\perp,BK} = \max_{B,K}\left\{u^2_{\perp,jk}\right\} \tag{6.29}$$

The inter-body velocity impact $u_{\perp,BK}$ is now defined as the highest among the particle impact velocities, while the relative inter-body distance is considered as the minimum among the corresponding inter-particle distances. In practise, $u_{\perp,BK}$ can be roughly, but more efficiently, estimated as the sum of the absolute values of the two body particles, whose interaction shows the highest relative velocity in the system.

One may now derive a proper definition for α_I, by equalling P_{BK} to P_{BK}':

$$\alpha_I = \left.\sum_K \frac{u^2_{\perp,BK}}{r_{per,BK}}\frac{m_B m_K}{m_B + m_K}\Gamma_{BK} \middle/ \sum_j\sum_k\left[\frac{u^2_{\perp,jk}}{r_{per,jk}}\frac{m_j m_k}{m_j + m_k}\Gamma_{jk}\left(1 - \frac{r_{par,jk}}{dx_s}\right)\right]\right. \tag{6.30}$$

In practise, the model prefers using the following approximated formulation to speed-up the simulations:

$$\alpha_I = \left.\sum_K \frac{1}{r_{per,BK}}\frac{m_B m_K}{m_B + m_K}\Gamma_{BK} \middle/ \sum_j\sum_k\left[\frac{1}{r_{per,jk}}\frac{m_j m_k}{m_j + m_k}\Gamma_{jk}\left(1 - \frac{r_{par,jk}}{dx_s}\right)\right]\right. \tag{6.31}$$

This is equivalent to considering the body impact velocity as a weighted average of the particle impact velocities.

At a first approximation, the normalizing factor α_I roughly represents the inverse of the fraction of the system mass which lies into the impingement zone. This mass should numerically represent the 2-body system during the impact. On the other hand, one cannot use (6.30) to model a body-body impact. In this case, for example, a definition for the direction of P_s' is required, but the direction of the relative distance between the two bodies does not avoid mass penetration. This would happen, for example, if two cubic bodies, very close to each other and with null barycentre velocities, began to rotate.

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):

$$\underline{P}_{BK*} = -\alpha_I \sum_j \frac{2u_{\perp,jK*}^2}{r_{per,jK*}} m_j \Gamma_{jK*} \underline{n}_{K*}, \qquad \alpha_I = \left. \frac{m_B}{r_{per,BK*}} \Gamma_{BK*} \middle/ \sum_j \left(\frac{m_j}{r_{per,jK*}} \Gamma_{jK*} \right) \right. \tag{6.32}$$

7. THE NUMERICAL MODEL: BED-LOAD TRANSPORT

This section describes a preliminary model for bed-load transport (Amicarelli & Agate, 2014, [4]; Sec.7.1) and its possible speed-up by means of a 2-interface 3D erosion scheme (Amicarelli & Agate, 2014, [4], Sec.7.2), which extends the main (1-interface) 2D erosion scheme of Manenti et al. (2012, [98], Sec.7.2).

This bed-load transport model is a SPH adaptation of the approximated model of Chauchat & Médale (2010, [30]) and is not consistent with the "packing limit" of the Kinetic Theory of Granular Flow (KTGF). A corrected and upgraded version of this preliminary model will be released with the following versions of SPHERA.

In this code version, the bed-load transport model can only be associated with the boundary treatment of the Semi-Analytic approach (SASPH).

7.1. Preliminary model for bed-load transport

The preliminary model for bed-load transport represents the dynamics of a mixture of pure liquid and (solid) granular material. In particular, mixture viscosity does not need any calibration.

The mixture density is defined as:

$$\rho = \rho_f \varepsilon_f + \rho_s \varepsilon_s \tag{7.1}$$

where ε represents the phase volume fraction and the subscripts "$_f$" and "$_s$" refers the fluid and solid phases, respectively. The volume equation reads:

$$\varepsilon_s + \varepsilon_f = 1 \tag{7.2}$$

Molecular diffusion is not represented as the model deals with suspensions (not solutions) and the relative velocity of phases is null by hypothesis.

The Weakly-Compressible approach ("WC") makes the mixture density to slightly vary. This allows estimating pressure depending on the density displacement from its reference value (barotropic equation of state).

The continuity equation assumes the following form:

$$\frac{d\rho}{dt} = -\rho \frac{\partial u_j}{\partial x_j} \tag{7.3}$$

Provided the boundary treatment of (Di Monaco et al., 2011, [39]), the SPH approximation of (7.3) reads:

$$\frac{d\rho_0}{dt} = \rho_0 \sum_b \left(u_{b,j} - u_{0,j}\right) \left.\frac{\partial W}{\partial x_j}\right|_b \omega_b + 2\rho_0 \int_{V_h'} \left[\left(\underline{u}_w - \underline{u}_0\right)\cdot \underline{n}\right] n_j \frac{\partial W}{\partial x_j} dx^3 \tag{7.4}$$

The velocity vector in the virtual region beyond the frontier is defined as in Di Monaco et al. (2011, [39]):

$$\underline{u}_{SA} = \underline{u}_{SA,T} + \underline{u}_{SA,n} \equiv \left(1 - c_s\right) u_{0,T} + \left[\left(2\underline{u}_w - \underline{u}_0\right)\cdot \underline{n}\right]\underline{n} \tag{7.5}$$

Consider Einstein dilute viscosity:

$$\mu_e = \mu\left(1 + \frac{5}{2}\phi_s\right) \tag{7.6}$$

The norm of the deviatoric (shear) stress tensor of the solid phase (in the bed-load transport layer) follows a generalized Mohr-Coulomb friction model:

$$\left\| \underline{\underline{\tau_s}} \right\| = \sigma' \quad tan\varphi \tag{7.7}$$

where φ is the internal friction angle and σ' is the vertical effective stress.
The norm of the mixture shear stresses assumes the following form:

$$\left\| \underline{\underline{\tau_m}} \right\| = \sigma' \quad tan\varphi + \mu_e \sqrt{I_2(e_{ij})} = \tau_c + \left(k\sqrt{I_2(e_{ij})} \right)^n, \tau_c = \left\| \underline{\underline{\tau_s}} \right\|, k = \mu_e, n = 1 \tag{7.8}$$

once provided the strain-rate tensor:

$$e_{ij} \equiv \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \tag{7.9}$$

Every SPH particle represents a volume of pure fluid ("fluid SPH particles", ε_s=0) or a mixture of saturated granular material ("mixture SPH particles").
In case porosity is unknown, the following values are adopted:

$$\varepsilon_f = \begin{cases} \varepsilon_{f,input}, & input-value \\ \varepsilon_{f,*}, & otherwise \end{cases}$$

$$\varepsilon_{f,*} = \begin{cases} \dfrac{\varepsilon_{f,\max} - \varepsilon_{f,\min}}{2} = 0.37, \varepsilon_{f,\max} = 0.48, \varepsilon_{f,\min} = 0.26, & uniform \quad size \\ 0.10, & otherwise \end{cases} \tag{7.10}$$

where the extreme values for uniform soils are derived from 3D analytic formulas.
Consider the definition of the apparent viscosity for a Bingham fluid (μ_a):

$$\mu_a \stackrel{\text{def}}{=} \frac{\left\| \underline{\underline{\tau}} \right\|}{\sqrt{I_2(2e_{ij})}} = \frac{\tau_c}{\sqrt{I_2(2e_{ij})}} + k\left(\sqrt{I_2(2e_{ij})} \right)^{n-1} \tag{7.11}$$

where τ_c is the critical shear stress, k consistency and n a characteristic exponent of the medium. After combining the above equations, one obtains the following expression for the apparent viscosity of the mixture:

$$\mu_m = \frac{\frac{\varepsilon_s}{\varepsilon_{s,max}} \sigma' \quad tan\varphi}{\sqrt{I_2(2e_{ij})}} + \mu_e \tag{7.12}$$

which is regularized by Chauchat & Médale (2010, [30]) as follows:

$$\mu_m = \frac{\varepsilon_s}{\varepsilon_{s,max}} \frac{\sigma' \quad tan\varphi_r}{\lambda \quad + \sqrt{I_2(e_{ij})}} + \mu_e, \qquad \lambda = \min\left(\lambda_*, \frac{\lambda_*}{Bn}\right), \qquad \lambda_* = 1.0 * 10^{-4} s^{-1},$$
$$Bn = \frac{\tau_c}{\mu_e}\left(\frac{H_s}{U_s}\right)^n \tag{7.13}$$

Bn is Bingham number, λ_* a characteristic constant, H_s and U_s the scale height and velocity, respectively. Eq. (7.13) can be approximated by assuming $\lambda = \lambda_*$. This allows reducing the computational costs and does not introduce any appreciable difference in the test cases investigated in Amicarelli & Agate (2014, SPHERIC, [4]).
The momentum equation for the mixture reads:

$$\frac{du_i}{dt} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} - \delta_{ij} g + \nu_m \frac{\partial^2 u_i}{\partial x_j^2}, \qquad \nu_m = \frac{\mu_m}{\rho} \tag{7.14}$$

The mixture (total) pressure is composed of the fluid pressure and the effective stress, whose formulation is very approximated:

$$p = p_f + \sigma' ,$$
$$\sigma' = (\gamma_m - \gamma_w)(z_{blt\_top} - z) = [\gamma_s(1-\eta) + \gamma_f \eta - \gamma_f](z_{blt\_top} - z)$$
$$= (\gamma_s - \gamma_f)(1-\eta)(z_{blt\_top} - z) \tag{7.15}$$

The subscript "$_{\text{blt-top}}$" represents the top of the bed-load transport layer.
The second invariant of the strain-rate tensor (free divergence flows) reads:

$$I_2(e_{ij}) = \frac{1}{2}\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial z}\right)^2\right] + \frac{1}{4}\left[\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)^2\right] \tag{7.16}$$

Provided the boundary treatment of (Di Monaco et al., 2011, [39]), the SPH approximation of (7.14) reads:

$$\left\langle \frac{du_i}{dt} \right\rangle_0 = -\delta_{i3} g + \frac{1}{\rho_0}\sum_b (p_b + p_0) \left.\frac{\partial W}{\partial x_i}\right|_b m_b + 2\frac{p_0}{\rho_0}\int_{V_h'} \left.\frac{\partial W}{\partial x_i}\right|_b dx^3 + +2\upsilon \sum_b \frac{m_b}{\rho_0 r_{0b}} (\underline{u}_b - \underline{u}_0) \left.\frac{\partial W}{\partial r}\right|_b +$$
$$+2\phi_s \upsilon (u_{w,i} - u_{0,i}) \cdot \left(\int_{V_h'} \frac{1}{r_{0w}} \frac{\partial W}{\partial r} dx^3 \right) - \upsilon_M \sum_b \frac{m_b}{\rho_0 r_{0b}^2} (\underline{u}_b - \underline{u}_0) \cdot (\underline{x}_b - \underline{x}_0) \left.\frac{\partial W}{\partial x_i}\right|_b - \upsilon_M (\underline{u}_{SA} - \underline{u}_0) \cdot \left(\int_{V_h'} \frac{1}{r_{0w}^2} (\underline{x} - \underline{x}_0) \frac{\partial W}{\partial x_i} dx^3 \right) \tag{7.17}$$

The artificial viscosity is always activated, both for approaching and separating particles (the latter configuration was not considered in Di Monaco et al., 2011, [39]):

$$\upsilon_M = \frac{\alpha h c}{\rho} \tag{7.18}$$

Renormalization (Randles & Libertsky, 1996, [137]) applies to the velocity derivatives in (7.16), only for 2D simulations:

$$\langle\frac{\partial f}{\partial x_i}\rangle \equiv \sum_b (f_b - f_0)\left(\underline{\underline{B}}_0 \cdot \underline{\nabla} W_b\right)_i \omega_b, \quad B_{0.ij}^{-1} \equiv \sum_b \left(\underline{x}_b - \underline{x}_0\right)_j \frac{\partial W}{\partial x_i} \omega_b \tag{7.19}$$

A barotropic equation of state (WC approach) for the mixture is linearized around the reference state (“$_{ref}$”):

$$p \cong c_{ref}^2 \left(\rho - \rho_{ref}\right) \tag{7.20}$$

The sound speed (c_{ref}) should be at least 10 times higher than the maximum velocity in the fluid (WC approach). It is sufficient to define a unique speed of sound for both mixture and pure water, as the maximum value resulting from considering all the numerical particles. The sound speed is computed by providing the bulk modulus as an input parameter for each medium:

$$K = \rho \frac{\partial \rho}{\partial p} = \rho c_{ref}^2 \tag{7.21}$$

The stability conditions for time integration are:

$$dt = CFL * min_0 \left\{ \frac{2h^2}{\upsilon_0}; \frac{2h}{c + \left|\underline{u}_0\right|}; \frac{2h^2}{\upsilon_{M,0}} \right\} \tag{7.22}$$

where CFL is the Courant-Friedrichs-Lewy number.

7.2. 2-interface 3D erosion criterion

A 2-interface 3D erosion criterion is implemented to speed-up the computational velocity of the model for bed-load transport (Sec.7.1), if the erosion is the only cause of mobilization of the solid grains. The erosion criterion aims to select those mixture particles, which needs the bed-load transport model to be applied.

The main erosion scheme is the 1-interface (“pure fluid - fixed bed”) 2D erosion criterion of Manenti et al. (2012, [98]), based on the formulation of Shields - van Rijn. Two modifications to this scheme are integrated: the extension to the third dimension and the treatment of a second interface (“bed-load transport layer - fixed bed”).

The erosion criterion refers to the interaction of a generic fixed mixture particle and the fluid flow above (pure fluid or mixture). Its reference parameters are represented by the closest mobile particle (of mixture or pure fluid) above the fixed particle. In any case, the interactions with the pure fluid are privileged, if available.

The formulation of van Rijn (1993, [172]) reads:

$$\vartheta_c = \begin{cases} 0.1, & Re_* < 1 \\ 0.010595 * \ln(Re_*) + \dfrac{0.110476}{Re_*} + 0.0027197, & 1 \le Re_* \le 500 \\ 0.068, & Re_* > 500 \end{cases} \tag{7.23}$$

where θ_c is Shields parameter and Re_* is the grain Reynolds number:

$$Re_* \stackrel{\text{def}}{=} \frac{k u_*}{\upsilon} \tag{7.24}$$

The assessment of the friction velocity (u_*) follows the procedure below.

If the reference height of the fluid (z) belongs to the Surface Neutral Boundary Layer (SNBL), the model computes the roughness coefficient z_0, according to the formula of Manenti et al. (2012, [98]) and those associated to the similarity theory or the SNBL:

$$z_0 = 0.11\frac{\upsilon}{u_*} + \frac{k}{30}, \quad u_* = \frac{k_\upsilon U}{ln\left(\frac{z}{z_0}\right)} \tag{7.25}$$

where k_v is von Karman constant and U is the flow velocity at the reference height.
If z refers to the SNBL, the model considers the velocity profile of the Sub-Viscous Layer, with a direct estimation of the friction velocity:

$$\mathrm{U} = \frac{zu_*^2}{\upsilon} \Rightarrow u_* = \sqrt{\frac{U\upsilon}{z}} \tag{7.26}$$

In this case, U can be smaller than u_*. This usually happens at the lower interface ("bed-load transport layer - fixed bed").
In synthesis, the model estimates u_* (by means of an iterative procedure if z refers to the SNBL -u_* depends on z_0, which is in turn function of u_*-), then Re_* and θ_c. Shield parameter is computed:

$$\vartheta \overset{\text{def}}{=} \frac{\tau}{(\gamma_s - \gamma)d}, \quad \tau = \rho u_*^2 \tag{7.27}$$

and compared with θ_c. The erosion criterion is satisfied if $\vartheta \geq \vartheta_c$
In practise, Shields criterion is derived under 1D stationary and uniform conditions, and does not explicitly depend on the friction angle. This is explicitly taken into account to quantify the effects of the fixed bed slope, as explained in the following.
The 2D erosion criterion for horizontal beds can be extended to 3D generic slopes, by means of the coefficient $k_{\beta\gamma}$, which is defined as follows:

$$\vartheta_{c,\beta\gamma} = k_{\beta\gamma}\vartheta_{c,00}, \quad 0 \leq k_{\beta\gamma} \leq k_{\beta 0} \tag{7.28}$$

$k_{\beta\gamma}$ is always non-negative and smaller than (or equal to) its 2D value $k_{\beta 0}$ (γ=0). In fact, if the slope angle transversal to the main flow direction (β) is not null, erosion is enhanced. Further, in the presence of a bed with a locally ascendant slope (β<0), $k_{\beta\gamma}$ can be higher than the unity. In this case, (7.28) can possibly provide a second non-physical solution, with $k_{\beta\gamma}$<1, which is not taken into account because it corresponds to a flow with an inverted direction.
The normal at the interface "bed-load transport - fixed bed" is defined by a means of a normalized SPH approximation of the relative distance between the mobile sub-domain and the generic SPH particle of the fixed bed:

$$\underline{n}_{int,0} = \frac{\sum_{bf}\left(\underline{x}_{bf} - \underline{x}_0\right)W_{bf}\omega_{bf}}{\left|\sum_{bf}\left(\underline{x}_{bf} - \underline{x}_0\right)W_{bf}\omega_{bf}\right|} \tag{7.29}$$

In the absence of a free surface, the normal is aligned with gravity, by definition.
The main slope angle quantifies the slope of the fixed bed in the direction of the main flow. Assuming that, close to the interface, the mixture velocity is parallel to the fixed bed, β only depends on the direction of the velocity vector of the closest particle (3D definition):

$$\beta = arcsin\left(-u_{b,3}\right) \tag{7.30}$$

In 2D, one could alternatively define β as function of the velocity direction or the interface normal. The latter assumption reduces the model errors and is used in 2D:

$$\beta = -\left[\frac{\pi}{2} - arcsin\left(n_{int,0,z}\right)\right] sign\left(u_{f,s,3}\right), \quad \underline{u}_{f,s} = \underline{u}_f - \underline{u}_f \cdot \underline{n}_{int} \tag{7.31}$$

The transversal slope angle γ is defined as:

$$\gamma = arcsin\left|n_{2,z}\right|, \quad \underline{n}_2 = \underline{n}_{int} \wedge \underline{u}_f \tag{7.32}$$

The unity vector $\underline{n}_2$ represents the bi-normal to the fluid particle trajectory and is independent on the sign of γ.
The value of $k_{\beta\gamma}$ is a solution of the quadratic equation of Seminara et al. (2002, [144]):

$$\begin{gathered}
ak_{\beta\gamma}^2 + bk_{\beta\gamma} + c = 0 \Rightarrow k_{\beta\gamma} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \\
a \stackrel{\text{def}}{=} (1 - \Delta), b \stackrel{\text{def}}{=} 2\left\{\frac{\Delta}{\sqrt{1 + \tan^2\beta + \tan^2\gamma}} + \frac{\sin\beta}{\tan\varphi}\right\}, \\
c \stackrel{\text{def}}{=} \frac{1 + \Delta}{1 + \tan^2\beta + \tan^2\gamma}\left(-1 + \frac{\tan^2\beta + \tan^2\gamma}{\tan^2\varphi}\right) \\
\Delta \stackrel{\text{def}}{=} \frac{4}{3}\tan\varphi \frac{C_L}{C_D}\frac{u_* d_{50}}{k_v U(z - z_{int})}
\end{gathered} \tag{7.33}$$

In the presence of two admissible roots, the model chooses the closest to $k_{\beta 0}$, provided $k_{\beta\gamma} \leq k_{\beta 0}$; in the absence of roots, the model assumes $k_{\beta\gamma}=k_{\beta 0}$.
The drag coefficient C_D is approximated by the formula of Morrison (2013, [116]) for a fluid flow around a sphere:

$$C_D = \begin{cases} \dfrac{24}{Re} + \dfrac{\frac{2.6Re}{5}}{1 + \left(\frac{Re}{5}\right)^{1.52}} + \dfrac{0.411\left(\frac{Re}{263'000}\right)^{-7.94}}{1 + \left(\frac{Re}{263'000}\right)^{-8}} + \dfrac{Re^{0.8}}{461'000}, & 1.0 * 10^2 \leq Re \leq 1.0 * 10^6 \\ 1.0, & Re \leq 1.0 * 10^2 \\ 0.2, & 1.0 * 10^6 \leq Re \end{cases} \tag{7.34}$$

with C_D varying between 0.1 and 1. Reynolds number is here defined as follows:

$$Re \stackrel{\text{def}}{=} \frac{U_\infty d_{50}}{\nu} \tag{7.35}$$

with U_∞ equal to the absolute value of velocity at the closest particle and d_{50} representing the 50-th percentile of the particle-size distribution of the soil.
In this context, the lift is assumes the form:

$$F_L = \frac{4}{3}\rho C_L \frac{\pi D^3}{8} U \frac{u_*}{k_v(z - z_{int})} \qquad (7.36)$$

where z_{int} is the interface height. A formula for the lift coefficient is derived, by interpolating the experimental data of Seminara et al. (2002, [144]):

$$C_L = \begin{cases} (9.0 * 10^{-5}) Re^{0.882}, & 2.0 * 10^3 \leq Re \leq 1.75 * 10^4 \\ 0.07, & Re \leq 2.0 * 10^3 \\ 0.5, & 1.75 * 10^4 \leq Re \end{cases} \qquad (7.37)$$

with C_L varying between 0.07 and 0.5.
The mixture pressure of a generic fixed SPH particle is computed, after assuming hydrostatic conditions within the fixed bed:

$$p_0 = \mathrm{p}_b + (z_b - z_0)\gamma_m + \frac{1}{2}\rho U_{nor,b}^2, \qquad U_{nor} = \left|\left(\underline{u}_b - \underline{u}_0\right) \cdot \left(\underline{x}_b - x_0\right)\right| \qquad (7.38)$$

Provided the absence of a fixed bed along the vertical and the simultaneous presence of fixed particles (or frontiers) within the kernel support, the mixture SPH particle is held fixed.

8. THE NUMERICAL MODEL: DB-SPH METHOD FOR BOUNDARY TREATMENT

This section describes the "Discrete Boundary" (DB) - SPH method for boundary treatment (Amicarelli et al., 2013, [69]). Consider that the activation of the DB-SPH method also alters the balance equations in the internal domain (Sec.6.2), as described in the following sub-sections.

8.1. DB-SPH particle approximation and modifications of the balance equations

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

$$\left\langle \left. \frac{\partial f}{\partial x_i} \right| \right\rangle_{\underline{x}_0} = \sum_a f_a W_a n_{i,a} \omega_a - \sum_b f_b W_b^{'} \omega_b, \qquad W_b^{'} \equiv \left. \frac{\partial W}{\partial x_i} \right|_b \tag{8.1}$$

In (8.1), the volume integral in (6.2) is replaced with a summation over the fluid particles within the kernel support. The surface integral of the same equation is replaced with a summation over the wall surface elements "$_a$" intercepted by the kernel support volume (V_h). (8.1) is normalized by the integral Shepard coefficient (γ) to obtain this further definition:

$$\left\langle \left. \frac{\partial f}{\partial x_i} \right| \right\rangle_{\underline{x}_0} \equiv \sum_a (f_A) \frac{W_A}{\gamma_0} n_{i,a} \omega_a - \sum_b (f_b) \frac{W_b^{'}}{\gamma_0} \omega, \quad \gamma \equiv \int_{V_h} W dx^3 \tag{8.2}$$

γ varies as function of the involved computational particle "$_0$". Provided fixed time and position, γ represents a constant for a particle equation system because it does not depend on the neighbouring particles. Thus, the normalization of the kernel derivative is simply obtained dividing by γ. This normalization allows considering the truncated kernel support as if it were entire (in the continuum), but with non-spherical shape.

Eq. (8.2) is used to approximate the pressure gradient term of Euler momentum equation (Sec.6.2). In the absence of the semi-particles, defined by Ferrand et al. (2013, [49]) in 2D, the boundary terms of (8.2) seem too modest to avoid the penetration of fluid particles trough the solid frontiers, once (8.2) is applied to the fluid dynamics balance equations. This limit seems due to the characteristics of the kernel function and its derivative (SPH truncation errors). Thus, the present model adopt semi-particles, whose 3D definition is slightly different from the edge particles (semi-particles) of Ferrand et al. (2013, [49]).

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).

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

Every discrete surface element represents a portion of frontier with area dx_w^2 (3D) or length dx_w (in 2D). At the same position, a fluid semi-particle is located. The semi-particle volumes are smaller than the fluid particle volumes not to alter the spatial resolution. The semi-particle position is located on one side of the physical volume of the semi-particle. However, this position should be representative of the entire semi-particle volume. This implies that the maximum distance between

any edge of the semi-particle and its position should be smaller than $dx_w/2$. Provided this constraint, the semi-particle depth coefficient should be high enough to improve the model accuracy.
Normally, SPH models do not consider the free surface as a frontier of the fluid domain as the atmospheric pressure is usually null in the gaseous sub-domain and on the free surface itself. Here, the DB-SPH approximation (8.2) introduces the parameter $p_0(\neq 0)$ in the surface terms of the momentum equation. Formally, one should explicitly model the free surface by means of surface elements over which summing the pressure gradient boundary terms of (8.2). In any case, this complication does not seem necessary if pressure gradients keep small enough at the free surface. This shortcoming is common in SPH mono-phase modelling (using other boundary treatments), and its effects are normally considered negligible (even because pressure gradients are generally zero at the very free surface).
When activating the DB-SPH boundary treatment, density in the inner domain is estimated by means of a SPH particle approximation, which replaces the continuity equation (Ferrand et al., 2013, [49]):

$$\langle \rho \rangle_0 = \frac{\sum_{bs} \rho_{bs} W_{bs} \omega_{bs}}{\tilde{\gamma}_0} \tag{8.3}$$

where the kernel is normalized by a corrected estimation of the integral Shepard coefficient.
The following correction of γ avoids excessive SPH truncation errors at the free surface:

$$\tilde{\gamma} \equiv \begin{Bmatrix} \sigma, & \gamma \geq \sigma + \varepsilon \\ \gamma, & \gamma < \sigma + \varepsilon \end{Bmatrix}, \qquad \sigma \equiv \sum_{bs} W_{bs} \omega_{bs} = \sum_{bs} W_{bs} \frac{m_{bs}}{\rho_{bs}} \tag{8.4}$$

The integral Shepard coefficient is replaced with the discrete Shepard coefficient at the free surface, which is numerically defined where $\gamma \geq \sigma + \varepsilon$. 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, [49]), as synthesized by (8.5) and (8.6). Consider the Lagrangian derivative of γ:

$$\frac{d\gamma}{dt} \cong \sum_a W_a \underline{n}_a \cdot \left(\underline{u}_a - \underline{u}_0 \right) \omega_a, \qquad \frac{\partial W}{\partial t} = 0 \tag{8.5}$$

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:

$$\gamma_0 (t = 0) \cong \begin{Bmatrix} 1, & \min\{r_{0a}\} \geq 2h \\ \sigma_0 (t = 0), & \min\{r_{0a}\} < 2h \end{Bmatrix} \tag{8.6}$$

The integral Shepard coefficient γ is initialized, according to the following procedure.

1) Some fictitious fluid particles are inserted in the computational domain to cover all the truncated parts of the kernel supports in the fluid domain (e.g., the gaseous sub-domain in mono-phase simulations of free surface flows). The density of the fictitious particles is negligible with respect to the computed fluid densities. The fictitious particles are neighbours of the computational particles, close to the free surface. The "fictitious

neighbouring particles" define several air volumes, which are provided as input "fictitious fluid volumes".

2) The model computes the initial values of γ by means of the approximated values provided by the estimation of the discrete Shepard coefficient. Thanks to the fictitious particles (having the same characteristic length of the computational particles), the estimation of σ (and then of γ) is sufficiently accurate, as the kernel supports are never truncated by the free surface.
3) The "fictitious air particles" can be removed at the end of γ initialization.

8.2. 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, [100]), 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 are described by means of a first order 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 a first-order spatial reconstruction scheme around he computational particle (f alternatively refers to density and every velocity component):

$$f_{0a}^{L} \cong f_0 + \left\langle \underline{\nabla f} \right\rangle_0 \cdot \left\langle \underline{x}_a - \underline{x}_0 \right\rangle, \qquad u_{n,0a} = \underline{u}_{0a}^{L} \cdot \underline{n}_a \tag{8.7}$$

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 EOS (mono-phase formulation):

$$\rho_{0a}^{*} = \rho_{0a}^{L} + \left(u_{n,0a}^{L} - u_{n,a}\right)\frac{\rho_{0a}^{L}}{c_{0a}^{L}}, \qquad p_{0a} = c^2\left(\rho_{0a}^{*} - \rho_0\right) \tag{8.8}$$

So far, the model has estimated several values of pressure, 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:

$$p_a = \frac{\sum_a p_{0a}(W\omega)_a}{\sum_a (W\omega)_a} \tag{8.9}$$

8.3. Semi-particle volume

The volume of a semi-particle is defined in Amicarelli et al. (2013, [5]):

$$\omega_s = k_w k_d dx_w \omega_a \tag{8.10}$$

where k_d represents the semi-particle shape coefficient and k_w the semi-particle depth coefficient.
The exact assessment of the shape coefficient is not an easy task. However, some exact solutions for noticeable cases are evaluated, both in 2D and 3D, based on the hypothesis of uniform angles (in the same configuration) with the number of adjacent faces equal to D (number of the spatial

dimensions). From those exact values, Amicarelli & Agate (2015, [3]) derive this interpolating formula:

$$k_d = \begin{cases} \dfrac{\sum_i \alpha_i}{D\dfrac{\pi}{2}} \dfrac{D}{n_{af}} - 1 = \dfrac{\sum_i \alpha_i}{\dfrac{\pi}{2} n_{af}} - 1, & \sum_i \alpha_i > \dfrac{\pi}{2} n_{af} \\ 0, \quad \sum_i \alpha_i \le \dfrac{\pi}{2} n_{af} & \end{cases} \tag{8.11}$$

where n_{af} represents the number of adjacent elements actually detected and the subscript "$_i$" here represents the generic adjacent element. The angles α_i lie between a generic surface element and each of the adjacent elements. According to the adopted formalism, the model needs to add 180° at the original assessment, in case the angle between the element normal vectors varies between -90° and +90°. The reference formula for α_i reads:

$$\alpha_i = \begin{cases} \pi + ar\cos(\underline{\hat{n}}_1 \cdot \underline{\hat{n}}_2) sign[\underline{\hat{n}}_0 \cdot \underline{d}_{0b}], & \underline{\hat{n}}_0 \cdot \underline{d}_{0b} \neq 0 \\ \pi, \quad \underline{\hat{n}}_0 \cdot \underline{d}_{0b} = 0 & \end{cases} \tag{8.12}$$

8.4. DB-SPH inlet and outlet sections

The inlet and outlet sections are represented by special surface elements, which are characterized by the following parameters: position, normal vector, null area (or length), pressure. Inlet and outlet surface elements allow detecting the computational particles, which are selected to impose inlet and outlet boundary conditions. The model search these particles within an influence sphere of characteristic length $L_c/\sqrt{2}$, where L_c represents the size of the inlet/outlet section. This search is very fast, but approximated: the accuracy of this simplified procedure depends on the test case. Once the interested computational particles are found, Dirichlet boundary conditions are assigned in terms of pressure and/or velocity components.

The inlet section is also interested by the following procedure, which reduces the SPH truncation errors. The free surface in the inlet region is made wavy to optimize the distribution of the fluid particles. The characteristic wave length is dx/2. The displacements are always perpendicular to the inlet normal. Two pattern regularly alternate. A white noise, with amplitude of dx/10, is finally added to the particle positions.

9. THE NUMERICAL MODEL: TIME INTEGRATION SCHEMES (LEAPFROG, EULER, HEUN)

Time integration is ruled by a second-order Leapfrog scheme (refer to [176] for stability analysis and time integration schemes in SPH modelling), as described in Amicarelli et al. (2015, [6]) and Di Monaco et al. (2011, [39]):

$$
\begin{aligned}
& x_{i,0}\Big|_{t+dt} = x_{i,0}\Big|_{t} + u_{i,0}\Big|_{t+dt/2} dt, \qquad i = 1,2,3 \\
& u_{i,0}\Big|_{t+dt/2} = u_{i,0}\Big|_{t-dt/2} + \left\langle \frac{du_{i,0}}{dt} \right\rangle\Big|_{t} dt, \qquad i = 1,2,3 \\
& \rho_0\Big|_{t+dt} = \rho_0\Big|_{t} + \left\langle \frac{d\rho_0}{dt} \right\rangle\Big|_{t+dt/2} dt
\end{aligned}
\tag{9.1}
$$

Two alternative explicit Runge-Kutta time integration schemes are also implemented: Euler scheme (RK1; first order) and Heun scheme (RK2, second-order).
According to RK1, the generic parameter f is integrated as follows:

$$
f(t_0 + dt) = f(t_0) + f'(t_0) dt(t_0), \qquad \frac{df}{dt} \equiv f' \tag{9.2}
$$

The scheme above can be rearranged in the following form:

$$
f_{RK1,i+1} = f_i + f'_i \, dt_i \tag{9.3}
$$

where the subscripts here represent the time step ID.
RK2 assumes the following form:

$$
\begin{aligned}
f_{RK2,i+1} &= f_i + \frac{dt}{2}\{f'(t_i, f_i) + f'\{t_{i+1}, [f_i + f'(t_i, f_i) dt_i]\}\} = \\
&= f_i + \frac{dt}{2}\left[f'(t_i, f_i) + f'(t_{i+1}, f_{RK1,i+1})\right]
\end{aligned}
\tag{9.4}
$$

This 2-stage formulation implies 2 stages (sub-loops) for each time step. During the first stage, the temporary value $f_{RK1,i+1}$ is computed. During the second stage, the time step value $f_{RK2,i+1}$ is assessed. However, several procedures do not need a double loop (e.g., the neighbouring search algorithm, the estimation of the time step duration, the inlet/outlet section management, the result printing, the erosion criterion).

10. DEVELOPER GUIDE

10.1. SPHERA v.8.0: synthetic description of the program units

The following sub-sections briefly describe all the program units of SPHERA v.8.0, according to their reference folder.

10.1.1. Program units for the boundary conditions ("BC")

The folder "BC" contains all the program units for the boundary conditions of the inlet and outlet sections (Table 10.1).

10.1.2. Program units for the continuity equation

The folder "BE_mass" contains all the program units to compute the Right Hand Side (RHS) of the continuity equation and the procedures of "partial smoothing" for pressure (Table 10.2).

10.1.3. Program units for the momentum equation

The folder "BE_momentum" contains the program units to compute the RHS of the momentum equation and the procedures of "partial smoothing" for velocity (Table 10.3).

10.1.4. Program units for the transport of solid bodies in free surface flows

The folder "Body_Transport" contains the program units exclusively dedicated to the transport of solid bodies in free surface flows (Amicarelli et al., 2015, [6]; Table 10.4).

Program unit	**Synthetic Description**
CancelOutgoneParticles_2D	To count and delete the outgoing particles on boundaries of type "leve", "flow", "velo", "crit", "open".
CancelOutgoneParticles_3D	To count and delete the outgoing particles on boundaries of type "leve", "flow", "velo", "crit", "open". Deletion occurs in 2 different ways: a) If the particle belongs to a particle zone (maxzone) with the highest index (the only zone where both particle number reduction and increase are allowed), then the outgoing particle (npi) is replaced by the last particle (nag) in the particle array pg, and the total number of particle becomes nag=nag-1; simultaneously, the index of the last particle of the zone is changed (Partz(maxzone)%limit(2)); b) Otherwise, simply pg(npi)%cella = 0 (particle out of the domain boundaries).
FindFrame	It finds extremes of the rectangular frame which contains the boundary mib.
FindLine	Finds extremes of the rectangular frame which contains the boundary mib.
GenerateSourceParticles_2D	To generate new source particles to simulate inlet fluid flow (only in 2D and with one inlet section).
GenerateSourceParticles_3D	To generate new source particles at the inlet section (only in 3D and with one quadrilateral inlet section).
IsParticleInternal2D	To check whether a particle is internal to the 2D domain.
IsParticleInternal3D	To check whether a particle is internal to the 3D domain or not. It checks if point Px() is internal to the perimeter mib. It returns 'true' (positive check) or 'false'. The perimeter can be both convex or concave.
NormFix	Minor program unit.
NumberSectionPoints	Minor program unit.
PreSourceParticles_2D	To generate new source particles at the inlet section (only in 2D and with one inlet section).
PreSourceParticles_3D	To generate new source particles at the inlet section (only in 3D and with one quadrilateral inlet section).
VelLaw	To impose an input kinematics to particles.

Table 10.1. Program units for the boundary conditions of the inlet/outlet sections ("BC"; SPHERA v.8.0).

Program unit	Synthetic Description
CalcPre	Particle pressure estimation.
inter_EqCont_2D	To accumulate contributions for the 2D continuity equation. Computation of velocity gradients and the second invariant of the strain-rate tensor.
inter_EqCont_2D	To accumulate contributions for the 3D continuity equation. Computation of velocity gradients and the second invariant of the strain-rate tensor.
inter_SmoothPres	To calculate a corrective term for pressure.
PressureSmoothing_2D	Partial smoothing for pressure (Di Monaco et al., 2011), also with DB-SPH boundary treatment scheme.
PressureSmoothing_3D	Partial smoothing for pressure (Di Monaco et al., 2011), also with DB-SPH boundary treatment scheme.

Table 10.2. Program units for the continuity equation (“BE_mass”; SPHERA v.8.0).

Program unit	Synthetic Description
Diffumorris	Minor subroutine.
inter_EqMoto	Computation of the momentum equation RHS (with DB-SPH boundary treatment scheme, Shepard's coefficient and gravity are added at a later stage) and the energy equation RHS (this last equation is not validated).
inter_SmoothVelo_2D	To calculate a corrective term for velocity.
inter_SmoothVelo_3D	To calculate a corrective term for velocity.
viscomon	Monaghan (2005) artificial viscosity term. It is also active for separating particles. Volume viscosity term is neglected in the momentum equation.
viscomorris	Morris term in the momentum equation.

Table 10.3. Program units for the momentum equation (“BE_momentum”; SPHERA v.8.0).

Program unit	Synthetic Description
Body_dynamics_output	.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 (Amicarelli et al., 2015, CAF)
body_pressure_postpro	Post-processing for body particle pressure.
body_to_smoothing_pres	Contributions of body particles to pressure partial smoothing (Amicarelli et al., 2015, CAF)
body_to_smoothing_vel	Contributions of body particles to velocity partial smoothing (Amicarelli et al., 2015, CAF)
Gamma_boun	Interpolative function defined by Monaghan (2005) for boundary force particles (Amicarelli et al.,2015,CAF).
Input_Body_Dynamics	Input management for body transport in fluid flows.
RHS_body_dynamics	To estimate the RHS of the body dynamics equations (Amicarelli et al.,2015,CAF).

Table 10.4. Program units for the transport of solid bodies in free surface flows (“Body_Transport”; SPHERA v.8.0).

Program unit	Synthetic Description
mixture_viscosity	To compute the mixture viscosity of the bed-load transport layer.
Viscapp	Constitutive equation with tuninig parameters (validated in Manenti et al.,2012,JHE)

Table 10.5. Program units for the constitutive equation (“Constitutive_Equation”; SPHERA v.8.0).

10.1.5. Program units for the constitutive equation

The folder “Constitutive_Equation” contains the program units for the constitutive equation (Table 10.5).

10.1.6. Program units for the boundary treatment scheme DB-SPH

The folder "DB_SPH" contains those program units, which are exclusively dedicated to the boundary treatment scheme DB-SPH (Amicarelli et al., 2013, [5]; Table 10.6).

10.1.7. Program units for the erosion criterion

The folder "Erosion_Criterion" contains those program units, which are exclusively dedicated to the 2D erosion criterion of Manenti et al. (2012, [98]) and its further developments (Table 10.7).

10.1.8. Program units on geometry (i.e., analytic geometry, algebra, …)

The folder "Geometry" contains the program units dedicated to analytic geometry, algebra and coordinate changes (Table 10.8).

Program unit	**Synthetic Description**
adjacent_faces_isolated_points	Provided 2 adjacent triangular/quadrilateral faces, it finds at least 2 vertices not in common, at least one per face. They are ID_face1_iso and ID_face2_iso. In case the faces are not adjacent, then false_hyp=.true.
BC_wall_elements	Wall element density and pressure (Amicarelli et al., 2013, IJNME).
DBSPH_find_close_faces	Finding the adjacent surface elements of a given surface element, both using 3D -triangular elements- and 2D -quadrilateral raw elements- configurations (DB-SPH).
DBSPH_IC_surface_elements	Initialization of wall surface elements (Amicarelli et al., 2013, IJNME).
DBSPH_inlet_outlet	Impose boundary conditions at the inlet and outlet sections (DB-SPH boundary treatment scheme).
DBSPH_kinematics	Imposing input kinematics for the DB-SPH elements (linear interpolation of input data).
Gradients_to_MUSCL	0th-order consistency estimation of velocity and density gradients for the MUSCL reconstruction (to feed the Partial Linearized Riemann Solver; Amicarelli et al., 2013, IJNME).
Gradients_to_MUSCL_boundary	Estimation of the boundary terms for the MUSCL reconstruction scheme (DB-SPH), in case they are required in input.
Import_ply_surface_meshes	To import the surface meshes (generated by SnappyHexMesh -OpenFOAM-), as converted by Paraview into .ply files. This subroutine is mandatory and activated only for the DB-SPH boundary treatment scheme.
semi_particle_volumes	To compute the semi-particle shape coefficients and volumes.
viscomorris_wall_elements	Wall element contributions to Morris' viscosity term.
wall_elements_pp	Smoothing wall element values for post-processing. Post-processing the wall surface element values (provided a selected region). Post-processing the hydrodynamic normal force on DBSPH surface elements (provided a selected region). Post-processing the wall surface element values (provided selected element IDs).
wavy_inlet	To provide a very slightly wavy flow at the inlet section. Each particle layer is staggered by 0.5dx with respect to the previous and the following ones, which are instead aligned each other. This numerical feature reduces the SPH truncation errors at the DB-SPH inlet sections. A white noise is also added. (Amicarelli et al., 2013, IJNME).

Table 10.6. Program units for the boundary treatment scheme DB-SPH ("DB_SPH"; SPHERA v.8.0).

Program unit	**Synthetic Description**
compute_k_BetaGamma	To compute k_BetaGamma=teta_c/teta_c,00. k_BetaGamma is the ratio between Shields critical non-dimensional stress for a generic 3D slope (teta_c) and its analogous value defined by Shields diagram (teta_c,00) on flat bed.
fixed_bed_slope_limited	Forced deposition (or no erosion) for particles at least 2h below the fixed bed (as it is defined in the associated column) during the same time step: i.e. the maximum slope of the fixed bed is 2h/2h. This avoids eventual too fast propagation of erosion along the vertical (erosion is an interface phenomenon).

MohrC	Mohr-Coulomb 2D erosion criterion (Manenti et al., 2012, JHE). Shields erosion criterion works better (Manenti et al., 2012, JHE).
Shields	3D erosion criterion based on the formulation of both Shields-van Rijn 2D criterion and Seminara et al.(2002) 3D criterion. 2D Shields erosion criterion based on pure fluid - fixed bed interactions (Manenti et al., 2012, JHE). Extension for bed load transport layer - fixed bed interactions (Amicarelli et al., CAF, submitted). Extension to the third dimension (Amicarelli et al., CAF, submitted). k=3d_90 (Manenti et al., 2012, JHE; Amicarelli et al., CAF, submitted). Shields threshold for low Re* (Amicarelli et al., CAF, submitted).

Table 10.7. Program units for the erosion criterion ("Erosion_Criterion"; SPHERA v.8.0).

Program unit	**Synthetic Description**
area_quadrilateral	Computation of the area of a generic quadrilateral from the coordinates of its vertices.
area_triangle	Computation of the area of a generic triangle, provided the coordinates of its vertices.
dis_point_plane	Computation of the distance between a point and a plane.
distance_point_line_2D	Computation of the distance between a point and a plane.
distance_point_line_3D	Computation of the distance between a point and a line in 3D.
IsPointInternal	Checking wheather a point with local normal coordinates csi() is internal to a given face, whose code is fk (=1 triangle, =2 parallelogram).
line_plane_intersection	Computation of the intersection point, if unique, between a line and a plane.
LocalNormalCoordinates	Given the local coordinates PX(1 to 2) of a point P laying on the plane of the boundary face nf, the procedure assigns to csi(1 to 3) the normal coordinates of the point Q corresponding to P in the inverse linear tranformation.
Matrix_Inversion_2x2	Computation of the inverse (inv) of a provided 2x2 matrix (mat).
Matrix_Inversion_3x3	Computation of the inverse (inv) of a provided 3x3 matrix (mat).
MatrixProduct	Returning in CC the product between matrices AA and BB. nr: number of rows of AA and CC. nc: number of columns of BB and CC. nrc: number of columns of AA = number of rows of BB.
MatrixTransposition	Returns in AAT(n,m) the transposed matrix of AA(m, n).
point_inout_polygone	Test to evaluate if a point lies inside or strictly outside a polygone (a triangle or a quadrilateral).
quadratic_equation	To solve a quadratic equation.
reference_system_change	Transformation of coordinates, expressed in a new reference system.
three_plane_intersection	Computation of the intersection of 3 planes.
Vector_Product	To return in ww the cross product of vectors uu and vv.
vector_rotation	Rotation of a given vector, provided the vector of the rotation angles (3D).

Table 10.8. Program units on Geometry ("Geometry"; SPHERA v.8.0).

10.1.9. Program units for the initial conditions (IC)

The folder "IC" contains the program units on the management on the initial conditions (Table 10.9).

10.1.10. Draft program units for the turbulent dispersion of granular material

For sake of completeness with respect to the previous versions of the code, the folder "Interface_dispersion" contains the draft program unit "inter_CoefDif". This computes a corrective term for particle velocity around the interface "mixture - pure fluid".

10.1.11. Program units for the main algorithms

The folder "Main_algorithm" contains the main program ("main") and the program units for the main code algorithms (both in 2D and 3D), the memory management and the Leapfrog time integration scheme (Table 10.10).

Program unit	Synthetic Description
GeneratePart	Particle positions (initial conditions).
initialization_fixed_granular_particle	To initialize the most of the fixed SPH mixture particles (bed-load transport).
SetParticleParameters	Setting initial particle parameters.
SetParticles	Particle coordinates (initial conditions).
SubCalcPreIdro	Hydrostatic pressure profiles (in case they are imposed as initial conditions).

Table 10.9. Program units for the initial conditions (“IC”; SPHERA v.8.0).

Program unit	Synthetic Description
Gest_Dealloc	Deallocations.
Gest_Trans	Introductory procedure for the main algorithm.
Loop_Irre_2D	2D main algorithm.
Loop_Irre_3D	3D main algorithm.
sphera	Main program unit.

Table 10.10. Program units for the main algorithms (“Main_algorithm”; SPHERA v.8.0).

10.1.12. Modules

The folder “Modules” contains the Fortran modules of SPHERA v.8.0. (Table 10.11).

10.1.13. Program units for the neighbouring search, the smoothing operators and the interface detection.

The folder “Neighbouring_Search” contains the program units for the neighbouring search, the kernel function and derivatives and the detection of the interfaces for the bed-load transport (Table 10.12).

10.1.14. Program units for post-processing

The folder “Post_processing” contains the program units to post-process the code results (Table 10.13). The main output files report the following parameters:

- flow rate hydrographs at the flow rate monitoring sections;
- 2D fields of the maximum values of the specific flow rate and the free surface height;
- time evolution of the interfaces of the bed-load transport model;
- time evolution of the main fluid dynamics variables (pressure and velocity) along the monitoring lines and points;
- hydrographs of the free surface height along the monitoring points;
- application log of SPHERA;
- 2D fields of the main fluid dynamics and SPH variables (“.vtu” and “.pvd” file formats) for Paraview (graphic FOSS) visualization;
- frontier geometry for the boundary treatment SA-SPH (“.vtk” format for Paraview);
- output files of the boundary treatment scheme DB-SPH (ref.: folder “DB_SPH”);
- output files on the transport of solid bodies in free surface flows (ref.: folder “Body_dynamics”).

10.1.15. Program units for pre-processing

The folder “Pre_processing” cntains the program units (Table 10.14) to pro-process the input files of SPHERA, which are:

- main input file (“.inp” format is defined in SPHERA v.8.0; user-defined name);
- file list for the DB-SPH surface meshes (“surface_mesh_list.inp”);
- ensemble of the files of the DB-SPH surface meshes (“.ply” format), which can be generated by means of SnappyHexMesh (FOSS mesh generator, OpenCFD Ltd) or Paraview.

Program unit	Synthetic Description
Dynamic_allocation_module	Module to define dynamically allocated variables.
Hybrid_allocation_module	Module to define derived types of both dynamically and statically allocated variables. (Di Monaco et al., 2011, EACFM; Manenti et al., 2012; JHE; Amicarelli et al., 2013, IJNME; Amicarelli et al., 2015, CAF).
I_O_diagnostic_module	To provide global interfaces to the subroutine diagnostic.
I_O_ENG_module	Minor module
I_O_file_module	Module for I/O.
I_O_ITA_module	Minor module
I_O_language_module	Minor module
SA_SPH_module	Module for the semi-analytic approach (boundary treatment scheme) of Di Monaco et al. (2011, EACFM).
Static_allocation_module	Module to define global (and statically allocated) variable.
Time_module	Module for time recording.

Table 10.11. Fortran modules (“Modules”; SPHERA v.8.0).

Program unit	Synthetic Description
CalcVarLength	Neighbouring search (pre-conditioned dynamic vector), relative positions, kernel functions/derivatives, Shepard's coefficient, position of the fluid-sediment interfaces along each background grid column.
CellIndices	To return the indices (i,j,k) of the cell (nc) in a 3D domain with ni*nj*nk cells.
CellNumber	To return the ID of the cell of indices (i,j,k).
CreaGrid	To create the background positioning grid.
InterFix	Minor program unit
OrdGrid1	Ordering the numerical elements on the background positioning grid.
ParticleCellNumber	To return the ID of the grid cell where particle np is located. If particle is outside of the grid, it returns -1.
SearchforParticleZone_3D	It returns in "partizone" the highest index of wet cells. In case no cell is wet, "partzone = sourzone" ("sourzone"is the inlet section cell).
w	kernel function

Table 10.12. Program units for the neighbouring search, the smoothing operators and the interface detection (“Neighbouring_Search”; SPHERA v.8.0).

Program unit	Synthetic Description
calc_pelo	Post-processing to write the free surface height.
CalcVarp	To calculate physical quantities at a monitoring point.
CreateSectionPoints	Minor program unit
GetVarPart	Getting particle values.
Memo_Ctl	Post-processing for monitoring lines and points.
Memo_Results	To write detailed results for restart. Not recommended.
Print_Results	Post-processing for the log file.
result_converter	Post-processing for .vtu (fluid dynamics parameters) and .vtk (geometry) files for Paraview.
s_ctime	Minor program unit
start_and_stop	Time recording.
sub_Q_sections	Writing flow rate at monitoring sections provided in input for the flow rate (only in 3D).
Update_Zmax_at_grid_vert_columns	Updating the 2D array of the maximum values of the fluid particle height, for each grid columns (only in 3D). Printing the 2D field of the water depth (current time step), according to the output frequency chosen in the input file (only in 3D). Printing the 2D fields of the

	specific flow rate components (current time step), at the same frequency of the water depth (only in 3D).
write_Granular_flows_interfaces	Post-processing the interfaces for bed-load transport phenomena.
write_h_max	To compute and write the 2D array of the maximum values of the water depth, at the nodes of the Cartesian topography, provided as input data (only in 3D). Same task for the 2D field of the maximum (over time) specific flow rates.

Table 10.13. Program units for post-processing (“Post_processing”; SPHERA v.8.0).

Program unit	**Synthetic Description**
defcolpartzero	On the particle colours for visualization purposes.
Diagnostic	Diagnostic (error) messages.
Gest_Input	Input check and management.
Init_Arrays	Minor program unit
ModifyFaces	To generate triangles from quadrilaterals (partitioning along the shortest diagonal)
ReadBedLoadTransport	Reading input data for bed-load transport.
ReadBodyDynamics	Reading input data for body trasnport in fluid flows (Amicarelli et al., 2015, CAF).
ReadCheck	Minor program unit
ReadDBSPH	Reading input data for the DB-SPH boundary treatment scheme (Amicarelli et al., 2013, IJNME).
ReadInput	Reading input data.
ReadInputBoundaries	Reading input data for the boundary treatment scheme SA-SPH (semi-analytic approach; Di Monaco et al., 2011, EACFM).
ReadInputControlLines	Reading monitoring lines.
ReadInputControlPoints	Reading monitoring points.
ReadInputControlSections	Reading control sections (not valid for the flow rate)
ReadInputDomain	Minor program unit
ReadInputDrawOptions	Minor program unit
ReadInputExternalFile	Minor program unit
ReadInputFaces	Minor program unit
ReadInputGeneralPhysical	Minor program unit
ReadInputLines	Minor program unit
ReadInputMedium	Minor program unit
ReadInputOutputRegulation	Minor program unit
ReadInputParticlesData	Minor program unit
ReadInputRestart	Minor program unit
ReadInputRunParameters	Minor program unit
ReadInputTitle	Minor program unit
ReadInputVertices	Minor program unit
ReadRiga	Minor program unit
ReadSectionFlowRate	Input management for the flow rate monitoring sections.

Table 10.14. Program units for pre-processing (“Pre_processing”; SPHERA v.8.0).

Program unit	**Synthetic Description**
AddBoundaryContribution_to_CE2D	To compute boundary terms for the 2D continuity equation (rodivV). Equation refers to particle npi. It performs implicit computation of gradPsuro. (Di Monaco et al., 2011, EACFM).
AddBoundaryContribution_to_CE3D	To compute boundary terms for the 3D continuity equation (rodivV). Equation refers to particle npi. It performs implicit computation of gradPsuro. (Di Monaco et al., 2011, EACFM).
AddBoundaryContributions_to_ME2D	To compute boundary terms for the 2D momentum equation (gradPsuro,

	ViscoF). Equations refer to particle npi. (Di Monaco et al., 2011, EACFM).
AddBoundaryContributions_to_ME3D	To compute boundary terms for 3D momentum equation (gradPsuro, ViscoF). Equations refer to particle npi. It performs implicit computation of gradPsuro. (Di Monaco et al., 2011, EACFM).
AddElasticBoundaryReaction_2D	To add supplementariìy normal boundary reaction to support eventual insufficient pressure gradient boundary term. In case of few neighbouring particles and presence of normal component of mass force (gravity). The normal reaction is computed with the formula R=(c0^2/d) ln(zi/d) [for zi<d], stemming from the compressible reaction of the fluid, where: c0^2 = E/ro0 is the square of the sound speed within the fluid; zi is the distance of the particle Pi from the boundary face; d is a reference distance from which the reaction is added. Check that the elastic boundary reaction never works. To compute the boundary integral IntWdS (Di Monaco et al., 2011, EACFM).
AddElasticBoundaryReaction_3D	To add supplementary normal boundary reaction to support eventual insufficient pressure gradient boundary term. In case of few neighbouring particles and presence of normal component of mass force (gravity). The normal reaction is computed with the formula R=(c0^2/d) ln(zi/d) [for zi<d], stemming from the compressible reaction of the fluid, where: c0^2 = E/ro0 is the square of the sound speed within the fluid; zi is the distance of the particle Pi from the boundary face; d is a reference distance from which the reaction is added. Check that the elastic boundary reaction never works. (Di Monaco et al., 2011, EACFM).
BoundaryMassForceMatrix2D	Generation of the generalised boundary mass force matrix RN, on the base of the cosine matrix T and the parameter Fi. (Di Monaco et al., 2011, EACFM)
BoundaryMassForceMatrix3D	Generation of the generalised boundary mass force matrix RN, on the base of the cosine matrix T and the parameter Fi. (Di Monaco et al., 2011, EACFM)
BoundaryPressureGradientMatrix3D	To generate the pressure gradient matrix RRP, based on the cosine matrix T and the parameter vector Psi. (Di Monaco et al., 2011, EACFM)
BoundaryReflectionMatrix2D	Generation of the generalised reflection matrix R, based on the cosine matrix T and the parameters PsiS and PsiN. (Di Monaco et al., 2011, EACFM)
BoundaryVolumeIntegrals2D	To compute the boundary volume integrals IntWdV. (Di Monaco et al., 2011, EACFM)
CompleteBoundaries3D	(Di Monaco et al., 2011, EACFM)
ComputeBoundaryDataTab	To calculate the array to store close boundaries and integrals. (Di Monaco et al., 2011, EACFM)
ComputeBoundaryIntegralTab	To compute local coordinates (x,y,z) of a grid of points, regularly distributed on the semisphere z<0 (radius = 2h), whose centre is the origin O of local axis. The semisphere will be superposed to the influence sphere (kernel support) of a generic particle near a plane boundary face, and oriented in such a way that the axis (x,y,z) coincide with the face local axes (r,s,n). In the first three columns of the array BoundaryIntegralTab() the coordinates (x,y,z) of each point are stored; in the forth column the relative d_alpha (portion of solid angle relative to the point, necessary for integrations) is stored. BITcols = 4. (Di Monaco et al., 2011, EACFM)
ComputeBoundaryVolumeIntegrals_P0	(Di Monaco et al., 2011, EACFM)
ComputeKernelTable	To pre-compute and store in kerneltab(0:ktrows,0:ktcols) the following values: kerneltab(0:ktrows, 0) = rob = rb/h kerneltab(0:ktrows, 1) = Int W* ro2 dro (from rob to 2) kerneltab(0:ktrows, 2) = Int dW*/dro ro dro (from rob to 2) kerneltab(0:ktrows, 3) = Int dW*/dro ro^2 dro (from rob to 2) kerneltab(0:ktrows, 4) = Int dW*/dro ro^3 dro (from rob to 2) (Di Monaco et al., 2011, EACFM)
ComputeSurfaceIntegral_WdS2D	Computing the surface integral of kernel W along the segments intercepted by the kernel support (radius=2h) of the particle i, whose local coordinates are xpi=LocXY(1,icbs) and ypi=LocXY(2,icbs), on the adjacent boundary side icbs. (Di Monaco et al., 2011, EACFM)

ComputeVolumeIntegral_WdV2D	Computing the integral of WdV extented to the volume delimited by the kernel support (radius=2h) of the particle i, whose local coordinates are xpi=LocXY(1,icbs) and ypi=LocXY(2,icbs), and the adjacent boundary side icbs. (Di Monaco et al., 2011, EACFM)
DefineBoundaryFaceGeometry3D	To define boundary faces from 3D geometry. (Di Monaco et al., 2011, EACFM)
DefineBoundarySideGeometry2D	Definition of the boundary sides. (Di Monaco et al., 2011, EACFM)
DefineBoundarySideRelativeAngles2D	Detection of the previous adjacent side and associated relative angle (for each boundary side). (Di Monaco et al., 2011, EACFM)
DefineLocalSystemVersors	To define the directional cosines of the local reference system. (Di Monaco et al., 2011, EACFM)
EvaluateBER_TimeStep	(Di Monaco et al., 2011, EACFM)
FindBoundaryConvexEdges3D	To look for possible edges with an associated convex geometry. Their geometrical data are saved in BoundaryConvexEdge as TyBoundaryConvexEdge. (Di Monaco et al., 2011, EACFM)
FindBoundaryIntersection2D	To find the intersection segment between the kernel support of particle i, whose local coordinates are xpi=LocXY(1,icbs) and ypi=LocXY(2,icbs), and the straight boundary side iside=Cloboside(icbs), which lies on the local x-axis and extends from x=0 to bsidelen = BoundarySide(iside)%Length. It returns: xpmin: minimum abscissa of intersected segment xpmax: maximum abscissa of intersected segment interlen: length of the intersected segment (Di Monaco et al., 2011, EACFM)
FindCloseBoundaryFaces3D	To finds the "close" boundary faces, i.e. those faces located at a distance from the particle npi smaller than or equal to 2h. It returns: Ncbf: number of close boundary faces Clobface(1 to Ncbf): list of close boundary faces LocX(1:SPACEDIM,Ncbf): local coordinates of particle npi with respect each boundary side The algorithm looks for the boundary faces intersected by the cell boxes of the reference frame located all around particle npi, and cancels the repeated ones. (Di Monaco et al., 2011, EACFM)
FindCloseBoundarySides2D	To finds the "close" boundary sides, i.e. those sited at a distance from particle npi<=2h. It returns: Ncbs: number of close boundary sides (= 0, 1, 2) Cloboside(1:Ncbs): list of close boundary sides LocXY(1:PLANEDIM,1:Ncbs): local coordinates of particle npi with respect each boundary side (vertex V1) (Di Monaco et al., 2011, EACFM)
GridCellBoundaryFacesIntersections3D	To find the boundary faces intercepted by each frame cell of the grid nc[1,NumCells]. In the generic row nc of the vector CFBFPointers(1 to NumCells,1 to 2), it sets: in the first column: the number of the intercepted faces in the second column: the pointer to CFBFVector, where the list of intercepted faces begins Searching is based on a principle of exclusion and is carried out in two phases: First phase: for every cell, it excludes (as possibly intercepted) the faces, whose vertices all lie in one of the semispaces (defined by the planes containing the cell faces), which do not include the cell itself. Second phase: for every remaining face, it verifies if all the 8 cell vertices belong to one of the semispaces defined by the plane containing the face. In the positive case, the face is excluded. (Di Monaco et al., 2011, EACFM)
InterpolateBoundaryIntegrals2D	Interpolation in table "BoundIntegralTab(:,:)", defined in module "SA_SPH_module", the values in columns "Colmn(nc), nc=1, Ncols" corresponding to the input value "x" to be interpolated, in turn, in column 0. It returns: Func(nc), nc=1, Ncols : values interpolated in columns Col(nc), nc=1, Ncols (Di Monaco et al., 2011, EACFM)

InterpolateTable	It interpolates values in the array "Table()" with "nrows" rows and "ncols" columns. Independent variables are in column 0 of Table(): nicols: number of columns of dependent variables to be interpolated icol(): list of columns of dependent variables to be interpolated ivalue(): list of the "nicols" interpolated values (Di Monaco et al., 2011, EACFM)
IWro2dro	Computes a SA-SPH definite integral (Di Monaco et al., 2011, EACFM)
J2Wro2	Computes a SA-SPH definite integral (Di Monaco et al., 2011, EACFM)
JdWsRn	Computes a SA-SPH definite integral (Di Monaco et al., 2011, EACFM)
SelectCloseBoundarySides2D	Selecting among the close boundary sides, those that really give contribution to the equations of particle 'npi'. It returns: IntNcbs: number of close boundary sides, which give contribution (= 0, 1, 2) Intboside(1:IntNcbs): list of close boundary sides, which give contribution IntLocXY(1:PLANEDIM,1:Ncbs): local coordinates of particle np with respect each boundary side, which gives contribution (Di Monaco et al., 2011, EACFM)
WIntegr	Computing a SA-SPH definite integral (Di Monaco et al., 2011, EACFM)

Table 10.15. Program units for the boundary treatment scheme SA-SPH ("SA_SPH", SPHERA v.8.0)

Program unit	Synthetic Description
time_integration_body_dynamics	Euler time integration for body transport in fluid flows.
Euler	Explicit RK1 time integration scheme (Euler scheme).
Heun	Heun scheme: explicit RK2 time integration scheme.
inidt2	Initial time step.
rundt2	Time step computation according to stability constraints (inertia terms, visosity term, interface diffusion -not recommended-). Plus. a special treatment for Monaghan artificial viscosity term and management of low-velocity SPH mixture particles for bed-laod transport phenomena.
Stoptime	Stopping time.
time_integration	Explicit Runge-Kutta time integration schemes

Table 10.16. Program units for time integration ("Time_integration"; SPHERA v.8.0).

10.1.16. Program units for the boundary treatment scheme SA-SPH

The folder "SA_SPH" contains the program units, which are exclusively dedicated to the boundary treatment scheme SA-SPH (Di Monaco et al., 2011, [39]; Table 10.15).

10.1.17. Program units for managing Fortran character variables

Three minor program units are implemented to manage Fortran character variables: "GetToken", "lcase" and "ltrim" (folder "Strings").

10.1.18. Program units for time integration

The folder "Time_integration" contains those program units, which concern RK1 and RK2 time integration schemes (Table 10.16).

10.2. Style formatting

SPHERA developers follow the basic rules on Fortran 95 coding, adhere as much as possible to SPHERA file format and the following style formatting rules:

1) Please use the subroutine labels at the beginning of each program unit (title and description) and of sub-section ("modules", "declarations", "explicit interfaces", "allocations", "initializations", "statements", "deallocations").
2) Please use Fortran 95 standard and portable procedures to be compiled with both gfortran and ifort.

3) A generic program unit has to be named as the associated file (without file extension) to have simpler dependencies in the makefile. As a consequence, one file per program unit is allowed and vice versa.
4) Please write since the first column of each line.
5) Please use 3 blank spaces for indentation.
6) Please use 1 blank space only before and after any mathematical operator in the Right Hand Side of each assignment and when a blank space is clearly convenient in terms of readability. Otherwise, blank spaces are used only for indentation (and within comments). For example, "endif" and "enddo" better replace "end if" and "end do". Further, no blank space is present between a procedure and its arguments (e.g. write(*,*)).
7) For readability and printability, do not write beyond column 80. Here the symbol "&" is put for a new line.
8) Please follow this variable order for declarations: parameters, "inout" variables, local variables, external functions. For each of the previous variable set, please following the following sub-order: scalars, 1D arrays, …, nD arrays. Provided the same dimensionality, variable declarations follow this "sub-sub-order": "logical", "integer", "double precision", "character", derived types.
9) A comment begins with "! <capitol letter>" (there is a blank space after "!").
10) Any logical expression is written within brackets (e.g., "(a==b).and.(c==d)").
11) Automatic indentation is allowed only with blank spaces instead of tabs (but the makefile).
12) No multiple statements on a line (do not use ";" as a statement separator).
13) Do not go to a new line with "&" under the section "declarations".
14) Keywords are written in lower case letters (e.g.: do,if,…).
15) Comments are written in UK English.

Please, use Microsoft Equation Editor to update the equations of this file or to add new equations.

11. USER GUIDE

SPHERA installation is straightforward (Sec.11.1), even because the executable files are already compiled (with ifort and gfortran, also in debug mode).
SPHERA GitHub repository contains a sequence of input files, whose associated test cases are either reported on International Journal papers or represent their analogous simplifications. Please refer to SPHERA main references (Sec.1), the numerical model (Secs.6,7,8,9) and the verbose template for SPHERA main input file (Sec.11.2). This template defines and comments all the input parameters. Finally, SPHERA v.8.0 validations are reported in Sec.11.3.

11.1. Installation

SPHERA source and executable files are distributed on a dedicated Git repository on GitHub ([151]). In case of need, do not hesitate to use SPHERA contact email address (Sec.1).
SPHERA executable files are released for Linux OS (compilers: both ifort and gfortran, with OpenMP libraries).
The only mandatory argument (in the command line) of the chosen executable file is the name of the main input file (without the format extension ".inp").

11.2. Commented template of the main input file of SPHERA v.8.0

Figure 11.1 reports the commented template of the main input file of SPHERA v.8.0.
The comments define all the input parameters and describe the meaning of their possible values. Further, suggested or default values are reported.

Test case	Reference for detailed descriptions
2D_erosional_dam_break_SPHERA_demo	(simple test case, rough resolution)
2jets_plate_DBSPH_high_res	Amicarelli et al. 2013 (IJNME)
2jets_plate_DBSPH_low_res	Amicarelli et al. 2013 (IJNME)
2jets_plate_SASPH_low_res	Amicarelli et al. 2013 (IJNME)
Archimede	(simple test case, rough resolution)
asymmetric_wedge_20deg_light	Amicarelli et al. 2015 (CAF)
asymmetric_wedge_20deg_medium	Amicarelli et al. 2015 (CAF)
body-body_impact_asymmetric	Amicarelli et al. 2015 (CAF)
body-body_impact_low_vel	Amicarelli et al. 2015 (CAF)
body-body_impact_symmetric	Amicarelli et al. 2015 (CAF)
body-boundary_impact	Amicarelli et al. 2015 (CAF)
body-boundary_impact_low_vel	Amicarelli et al. 2015 (CAF)
dam_break_2_bodies	Amicarelli et al. 2015 (CAF)
dam_break_2D_demo	(simple test case, rough resolution)
dam_break_multi-body	Amicarelli et al. 2015 (CAF)
jet_body-plate	Amicarelli et al. 2015 (CAF)
jet_plate_DBSPH	Amicarelli et al. 2013 (IJNME)
jet_plate_DBSPH_low_res	Amicarelli et al. 2013 (IJNME)
jet_plate_SASPH_low_res	Amicarelli et al. 2013 (IJNME)
symmetric_wedge_20deg_light	Amicarelli et al. 2015 (CAF)
symmetric_wedge_20deg_medium	Amicarelli et al. 2015 (CAF)
water_box_free_surface	(simple test case, rough resolution)
water_tank-body	(simple test case, rough resolution)

Table 11.1. Input files in SPHERA GitHub repository.

```
version.subversion   !
                 ! SPHERA main input file: template and comments
```



```
##### TITLE #####
title             ! title (of the test case, string)
##### END TITLE #####

##### DOMAIN #####
! Input parameters for spatial resolution and boundary treatment scheme
D BC_string   ! D(spatial dimensionality)=2(2D),3(3D); BC_string(Boundary
            ! treatment scheme)=semi(SA-SPH),dbsph(DB-SPH)
dx h/dx r     ! The third parameter ("r") is optional and provides a white noise
            ! to the IC particle positions
##### END DOMAIN #####

##### VERTICES #####
! Input parameters for the boundary vertices of the fluid domain
! The DB-SPH boundary treatment scheme requires the
! vertices of the parallelepiped domain as a "contour" and the "fictitious
! air reservoirs"
! The SA-SPH boundary treatment scheme requires the vertices of the wall
! frontiers
! In absence of a declared origin, the first vertex is fictitious, equal to the
! and does not belong to any boundary (only useful for Paraview)
1 Vertex_x Vertex_y Vertex_z            ! (first vertex data)
! ...                              ! (other vertices)
Last_vertex_ID Vertex_x Vertex_y Vertex_z ! (last vertex data)
##### END VERTICES #####

! If (D==2D): start
##### LINES #####
```

```
! 2D input parameters for the boundary lines of the fluid domain
! 2D boundary lines for wall frontiers, inlet/outlet sections, fluid reservoirs.
! In case of DB-SPH boundary treatment scheme, the code requires the lines of
! the parallelepiped domain as a "perimeter" and the "air reservoirs".
! In case of SA-SPH boundary treatment scheme, the code requires the
! lines of the boundaries
1 line_1_vertex_1 ... line_1_vertex last line_1_Boundary_ID
                                ! first line data
...                             ! other records
line_last_ID line_last_vertex_1 ... line_last_vertex_last line_last_Boundary_ID
##### END LINES #####
! If (D==2D): end
! If (D==3D): start
##### FACES #####
! 3D input parameters for the boundary faces of the fluid domain
! 3D boundary faces for wall frontiers, inlet/outlet sections, fluid reservoirs.
! In case of DB-SPH boundary treatment scheme, the code requires the faces of
! the parallelepiped domain as a "perimeter" and the "air reservoirs".
! In case of SA-SPH boundary treatment scheme, the code requires the
! the boundary faces
! Vertex list clockwise (normal vector exiting the frontier/domain;
! view from the semi-space of the normal vector): not the best convention
! The last vertex ID is 0 in case of triangular faces
! SA-SPH boundary normal vectors and reservoir face normal vectors point inward
! (clockwise list of points if looking from outside the fluid domain). For
! "perimeter" boundaries: the vertices have to be adjacent, but there is no rule
! about the vertex order (better anticlockwise).
1 face_1_vertex_1 ... face_1_vertex last face_1_Boundary_ID
                                ! first face data
...                             ! other records
face_last_ID face_last_vertex_1 ... face_last_vertex_last face_last_Boundary_ID
##### END FACES #####
! If (D==3D): end

##### BOUNDARIES #####
! Input parameters for the fluid domain boundaries delimited by
! lines(2D)/faces(3D)
! In case of DB-SPH boundary treatment scheme, the parallelepiped domain
! (mandatory) is formally represented by a fictitious SA-SPH frontier, which
! is only used to generate the background positioning grid.
! 1st boundary
Boundary_name             !
Boundary_ID               !
Boundary_type             ! Boundary_type = fixed(wall frontier),perimeter
                    ! (fluid reservoir),source(inlet section),open,
                    ! tapis (not recommended)
! If (Boundary_type=="fixed"): start
Shear_stress_coefficient    ! Shear_stress_coefficient=1.0(no-slip),
                    ! 0.(free-slip)
RGBColor                  !
! If (Boundary_type=="fixed"): end
```

```
! If (Boundary_type=="perimeter"): start
fluid_ID            !
colour_pattern colour_ID    ! colour_pattern=uniform,bends; colour_ID=009EA8
! if (motion_type=std): start
motion_type IC_velocity_x IC_velocity_y IC_velocity_z slip_condition
                    ! motion_type=std; slip_condition=0.0
! if (motion_type=std): end
! if (motion_type=law): start
motion_type n_records       ! motion_type=law; n_records(number of records for
                    ! the imposed kinematics)
time_1 u_1 v_1 w_1 1
...
time_n_records u_n_records v_n_records w_n_records n_records
                    ! (list of records for the imposed
                    ! 3D translational kinematics: time, vector
                    ! velocity, record_ID)
! if (motion_type=law): end
IC_pressure_type IC_pressure_value
                    ! IC_pressure_type=pa(uniform pressure),qp
                    ! (hydrostatic conditions),pl(hydrostatic pressure
                    ! based on the maximum level of an assigned fluid;
                    ! IC_pressure_value=(uniform pressure value for pa),
                    ! (free surface height for qp), (equivalent free
                    ! surface level of the on-going fluid for pl)
IC_reservoir_type Car_top_zone
                    ! IC_reservoir_type=1(vertices and faces),2(from
                    ! Cartesian topography); Car_top_zone = boundary ID
                    ! of underlying topography(influence only if
                    ! IC_reservoir_type==2)
! If (IC_reservoir_type==2): start
dx_CartTopog H_res          ! dx_CartTopog(spatial resolution of the Cartesian
                    ! topography); H_res(height of the reservoir free
                    ! surface)
ID_first_vertex ID_last_vertex
                    ! ID_first_vertex,ID_last_vertex(ID of the first and
                    ! and the last vertices of the reference topography)
n_circum nag_aux            ! n_circum(number of vertices circumscribing
                    ! the horizontal projection of the reservoir)=3,4;
                    ! nag_aux(rough overestimation of the number of
                    ! fluid particles in the reservoir)
circum_1_x circum_1_y       ! First point of the 2D figure circumscribing the
                    ! horizontal projection of the reservoir
...                 ! other point/s of the 2D figure above
circum_last_x circum_last_y ! last point of the 2D figure above
dam_zone_ID n_circum_dam
                    ! dam_zone_ID; dam_zone_n_vertices(number of
                    ! vertices of the 2D figure circumscribing the
                    ! horizontal projection of the
                    ! dam zone)=3,4
circum_dam_1_x circum_dam_1_y
                    ! First point of the 2D figure circumscribing the
```

```
                        ! horizontal projection of the dam zone
...                     ! other point/s of the 2D figure above
circum_dam_last_x circum_dam_last_y
                        ! last point of the 2D figure above
! If (IC_reservoir_type==2): end
! If (Boundary_type=="perimeter"): end
! If (Boundary_type=="open"): start
RGBColor
! If (Boundary_type=="open"): end
! If (Boundary_type=="source"): start
flowrate 0.             ! flowrate(inlet velocity * inlet area); 0.
pa IC_pressure          !
RGBColor
! If (Boundary_type=="source"): end

! ...                   ! other boundaries
! n-th boundary
! ...                   ! Data of the last boundary
##### END BOUNDARIES #####

##### DBSPH #####
! Input parameters for the DB-SPH boundary treatment scheme
dx_f/dx_w MUSCL_boundary_flag k_w   ! dx_f/dx_w(ratio between the fluid particle
                              ! size and the wall element size);
                              ! MUSCL_boundary_flag(logical flag to
                              ! activate boundary terms for MUSCL); k_w
                              ! (semi-particle depth coefficient)
n_monitor_points n_monitor_regions  ! n_monitor_points; n_monitor_regions=0,1(to
                              ! estimate the Force along x-direction)
! if (n_monitor_points>0): start
ID_wall_element_monitor_1 ... ID_wall_element_monitor_n
! if (n_monitor_points>0): end
! if (n_monitor_regions>0): start
xmin,xmax,ymin,xmax,zmin,zmax       ! (monitoring region vertices)
! if (n_monitor_regions>0): end
imposed_kinematics_records flag_in-built_monitors
                              ! imposed_kinematics_records(number of
                              ! records, which describe a possible imposed
                              ! kinematics;
                              ! flag_in-built_monitors(logical): flag for
                              ! in-built motion of control lines and
                              ! DB-SPH frontiers
! if (imposed_kinematics_records==1): start
time_1 velocity_x_1 velocity_x_2 velocity_x_3
! ...                         ! other possible  records
time_last velocity_x_last velocity_x_last velocity_x_last
                              ! (records for the imposed translational
                              ! kinematics to frontiers)
! if (imposed_kinematics_records==1): end
n_inlet n_outlet                    ! n_inlet(number of inlet sections)
                              ! n_outlet(number of outlet sections)
```

```
! if (n_inlet>1): start
x_inlet_1 y_inlet_1 z_inlet_1 n_x_inlet_1 n_y_inlet_1 n_z_inlet_1 velocity_x_inlet_1
velocity_y_inlet_1 velocity_z_inlet_1 L_inlet_1
...                       ! (other possible records)
x_inlet_last y_inlet_last z_inlet_last n_x_inlet_last n_y_inlet_last n_z_inlet_last
velocity_x_inlet_last velocity_y_inlet_last velocity_z_inlet_last L_inlet_last
                          ! inlet section data: position, normal,
                          ! velocity, length.
! if (n_inlet>1): end
! if (n_outlet>1): start
x_outlet_1 y_outlet_1 z_outlet_1 n_x_outlet_1 n_y_outlet_1 n_z_outlet_1 velocity_x_outlet_1
velocity_y_outlet_1 velocity_z_outlet_1 L_outlet_1
...                       ! (other possible records)
x_outlet_last y_outlet_last z_outlet_last n_x_outlet_last n_y_outlet_last n_z_outlet_last
L_outlet_last p_outlet_last
                          ! outlet section data: position, normal,
                          ! length, pressure
! if (n_outlet>1): end
##### END DBSPH #####

##### BED LOAD TRANSPORT #####
! Input parameters for bed-load transport (blt) scheme
erosion_criterion_ID ID_main_fluid ID granular
                          ! erosion_criterion_ID=0(no bed-load
                          ! transport),1(Shields-Seminara),2(Shields
                          ! without blt-fixed bed interactions),3
                          ! (Mohr-Coulomb, not recommended);
                          ! ID_main_fluid(medium of
                          ! the main fluid); ID granular
                          ! (medium of the blt layer)
! if (erosion_criterion_ID>0): start
velocity_fixed_bed erosion_flag    ! velocity_fixed_bed(velocity threshold
                          ! -e.g. equal to velocity scale/100- to
                          ! detect the fixed bed); erosion_flag=0
                          ! (activated far from fronts); 1(inactive),
                          ! 2(active everywhere)
viscosity_blt_formula deposition_at_frontiers Gamma_slope_flag
                          ! viscosity_blt_formula(in the bed-load
                          ! transport layer)=1(Chauchat-Médale 2010
                          ! CMAME),2(Chezy-like),3(diluted viscosity),
                          ! 4(lambda(Bn));deposition_at_frontiers=1
                          ! (imposed),0(not imposed); Gamma_slope_flag
                          ! =1(Gamma slope angle computed),0(null)
n_monitor_lines dt_out erosion_convergence_criterion n_max_iterations
                          ! n_monitor_lines(number of monitoring lines
                          ! aligned with x- or y-axis); dt_out(writing
                          ! time step); erosion_convergence_criterion
                          ! (convergence criterion for the erosion
                          ! criterion); n_max_iterations(maximum
                          ! number of iterations for the erosion
                          ! criterion)
```

```
Chezy_friction_coefficient          ! Chezy_friction_coefficient(default=0.005)
x_min_dt x_max_dt
y_min_dt y_max_dt
z_min_dt z_max_dt                   ! Vertices of the parallelepiped, within
                                    ! which the mixture particles can influence
                                    ! the time step estimation
line_ID                             ! monitoring line ID for blt
x_line y_line                       ! monitoring line is defined by variable or
                                    ! fixed (-999.) x- and y-coordinates
! if (erosion_criterion_ID>0): end
##### end BED LOAD TRANSPORT #####

##### medium #####
! Input parameters for the fluids
fluid_type                          ! fluid_type=liquid,granular
fluid_ID                            !
! If (fluid_type==liquid): start
density bulk_modulus
! If (fluid_type==liquid): end
! If (fluid_type==granular): start
solid_phase_density solid_phase_bulk_modulus
                                    !
! If (fluid_type==granular): end
Monaghan_alpha Monaghan_beta        ! Monaghan alpha (artificial viscosity),
                                    ! Monaghan beta (=0, artificial viscosity)
diffusion_coefficient settling velocity coefficient
                                    ! null recommended values (i.e. inactive
                                    ! parameters)
0. 0. 0.
! If (fluid_type==liquid): start
dynamic_viscosity                   !
roughness_coefficient               ! null recommended values (i.e. inactive
                                    ! parameter)
! If (fluid_type==liquid): end
! if ((fluid_type==granular).and.(erosion_criterion_ID==1)): start
delta                               ! delta(internal friction angle in degrees,
                                    ! even if the code works in radians)
effective_porosity d_50 d_90        !
! if ((fluid_type==granular).and.(erosion_criterion_ID==1)): end
! if ((fluid_type==granular).and.(erosion_criterion_ID>1)): start
! Alternative to the reference blt scheme
cohesion viscosity_max tuned_viscosity
                                    ! cohesion; viscosity_max,tuned_viscosity (
                                    ! tuning parameters for viscosity)
delta                               ! delta(internal friction angle in degrees,
                                    ! even if the code works in radians)
roughness_coefficient d_50 erosion_model
                                    ! roughness_coefficient; d_50; erosion_model
                                    ! =shields,mohr
max_step_still                      ! max_step_still(number of time steps during
                                    ! which mixture particles are kept still)
```

```
! if ((fluid_type==granular).and.(erosion_criterion_ID>1)): end
##### end medium #####

##### BODY DYNAMICS #####
! Input parameters for the scheme on body transport in fluid flows
n_bodies dx/dx_body imping_body_grav
                              ! n_bodies(number of transported solid
                              ! bodies); dx/dx_body(ratio between fluid
                              ! particle size and body particle size);
                              ! imping_body_grav=0(gravity always
                              ! active, recommended value),1(gravity
                              ! inactive until the first impact
                              ! body-fluid)
! if (n_bodies>0): start
ID_first_body n_elem          ! ID_first_body=1; n_elem(number of
                              ! elements of the body)
body_mass                     !
pos_CM_x pos_CM_y pos_CM_z    ! pos_CM(position of the centre of mass at
                              ! t=0)
Ic_flag                       ! Ic_flag=0,1(mass moment of inertia is
                              ! imposed)
! if(Ic_flag==1): start
Ic(1,1) Ic(1,2) Ic(1,3)       !
Ic(2,1) Ic(2,2) Ic(2,3)       !
Ic(3,1) Ic(3,2) Ic(3,3)       !
! if(Ic_flag==1): end
alfa_x alfa_y alfa_z          ! alfa(rotation angles of the body axis with
                              ! respect to the reference system axis at
                              ! t=0)
pos_rotC_x pos_rotC_y pos_rotC_z ! pos_rotC(centre of rotation just to
                              ! configure the initial orientation in the
                              ! global reference system)
vel_CM_x vel_CM_y vel_CM_z    ! vel_CM(velocity of the centre of mass at
                              ! t=0)
omega_x omega_y omega_z       ! omega(angular velocity of the body at t=0)
imposed_kinematics_flag n_records ! imposed_kinematics_flag=0,1(kinematics is
                              ! imposed); n_records(number of records,
                              ! which desrcibe the imposed kinematics)
first_ID_element              ! first_ID_element=1(of body 1)
L_x L_y L_z                   ! L_x,L_y,L_z(side lengths of the element)
pos_CM_elem_x pos_CM_elem_y pos_CM_elem_z
                              ! pos_CM_elem(position of the centre of mass
                              ! of the element at t=0)
alfa_elem_x alfa_elem_y alfa_elem_z ! alfa_elem(rotation angles of the body axis
                              ! with respect to the reference system axis
                              ! at t=0)
face_xmin_flag face_xmax_flag face_ymin_flag face_ymax_flag face_zmin_flag face_zmax_flag
                              ! (integer flags to activate the normal
                              ! vectors of surface body particles only
                              ! if face_..._flag=1; x/y/z_min/max
                              ! indicates the 6 faces of the element
```

```
                           ! -parallelepiped-)
xmin xmax ymin ymax zmin zmax      ! (spatial limits -in the global reference
                           ! system before the initial rotation- to
                           ! deactivate particle masses if((x>=xmin).
                           ! or.(x<=xmax).or.(y>=ymin).or.(y<=ymax).or.
                           ! (z>=zmin).or(z<=zmax)) foor boolean
                           ! operations on elements/body)
...                        ! (other element records)
...                        ! (last element record)
!
...                        ! (other body records)
!
...                        ! (last body record)
! if (n_bodies>0): end
##### end BODY DYNAMICS #####

##### RUN PARAMETERS #####
! Input parameters for time integration, partial smoothing and memory management
final_time final_time_step        ! (the run stops when reaching either the
                           ! final time or the final time step)
CFL Leapfrog_flag scheme_order factor dt_alfa_Mon
                           ! CFL, Leapfrog_flag=1(Leapfrog time
                           ! integration scheme),0(explicit RK time
                           ! integration schemes); scheme_order(time
                           ! integration scheme order, but for Leapfrog
                           ! is "1"); factor(=0., weighting factor to
                           ! estimate dt); dt_alfa_Mon(logical flag
                           ! making Monaghan artificial viscosity
                           ! coefficient to influence dt)
teta_p, teta_u var                ! teta_p,teta_u(coefficients for partial
                           ! smoothing of pressure and velocity); var=A
COEFNMAXPARTI COEFNMAXPARTJ body_part_reorder
                           ! COEFNMAXPARTI:max0(max number of fluid
                           ! particles)=COEFNMAXPARTI*nag;
                           ! COEFNMAXPARTJ:maxb(max number of
                           ! neighbours)=COEFNMAXPARTJ*(4h/dx)^D;
                           ! body_part_reorder(DB-SPH)=0(fixed
                           ! frontiers),1(mobile frontiers)
MAXCLOSEBOUNDFACES MAXNUMCONVEXEDGES
                           ! MAXCLOSEBOUNDFACES(max number of
                           ! neighbouring boundary face per fluid
                           ! particle); MAXNUMCONVEXEDGES(max number
                           ! of edges)
GCBFVecDim                        ! GCBFVecDim(rough overestimation of the
                           ! number of Grid Cell - Boundary Face
                           ! intersections  (SA-SPH)
density_thresholds_flag           ! density_thresholds_flag=0(default, no
                           ! density limiters),1(density limiters for
                           ! debug)
##### end RUN PARAMETERS #####
```

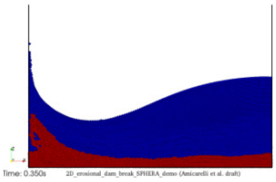
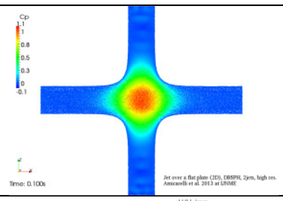
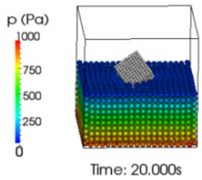
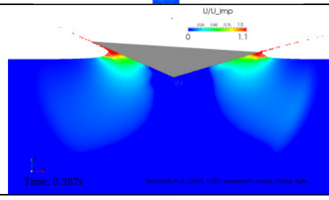
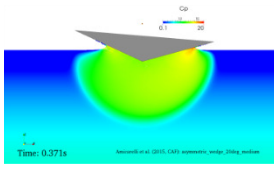
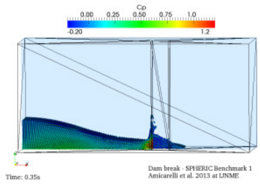
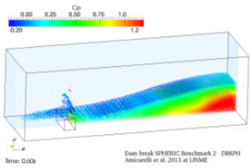
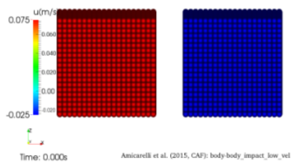
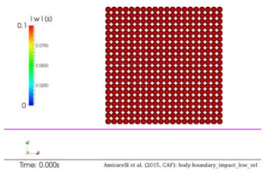
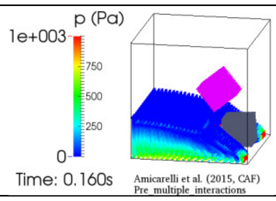
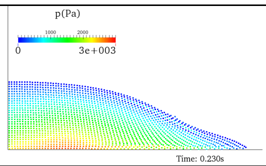
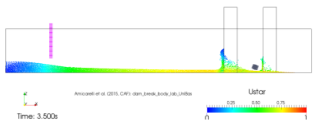
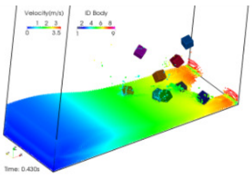
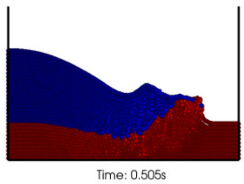
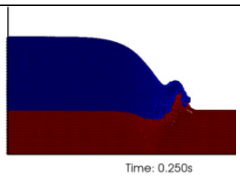
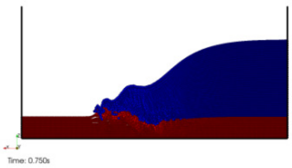
```
##### general physical properties #####
! Input parameters for gravity and reference pressure
! 3D case: start
gravity_acceleration_x gravity_acceleration_y gravity_acceleration_z
! 3D case: stop
! 2D case: start
gravity_acceleration_x gravity_acceleration_z
! 2D case: stop
reference_pressure              !
##### end general physical properties #####

!##### restart #####
! (this section is not active)
!##### end restart #####

##### output regulation #####
! Post-processing parameters for .txt files. The first two words (and the
! possible fourth) of each line are keywords.
results time dt_out             ! dt_out(writing time step)
restart time dt_restart         ! dt_restart(=99999., restart time step)
print partial log_file_frequency    ! log_file_frequency (log file writing time
                                ! step in terms of time step number)
control time dt_out_mon         ! dt_out_mon(writing time step for
                                ! monitoring elements)
level time dt_out_FS medium fluid_ID
                                ! dt_out_FS(writing time step for free
                                ! surface post-processing); fluid_ID
! if (IC_source_type==2): start
depth dt_out dt_out_depth       ! dt_out_depth(writing time step for 2D
                                ! fields of water depth (h) and specific
                                ! flow rate components (q_x=u_m*h,q_y=v_m*h))
! if (IC_source_type==2): end
##### end output regulation #####

##### draw options #####
! Post-processing parameters for Paraview file formats. The first two words
! of each line are keywords.
vtkconverter any dt_out_PV      ! dt_out_PV(writing time step for Paraview
                                ! .vtu files)
##### end draw options #####

##### control points #####
! Input parameters for monitoring points
x_monitor_point_1 y_monitor_point_1 z_monitor_point_1
...                             ! (other monitoring points)
x_monitor_point_n y_monitor_point_n z_monitor_point_n
##### end control points #####

##### control lines #####
! Input parameters for monitoring lines
line_1_label
```

```
! if(D==2D): start
edge_1_line_1_x edge_1_line_1_z
edge_2_line_1_x edge_2_line_1_z
! if(D==2D): end
! if(D==3D): start
edge_1_line_1_x edge_1_line_1_y edge_1_line_1_z
edge_2_line_1_x edge_1_line_1_y edge_2_line_1_z
! if(D==3D): end
line_1_number_of_discretization_points
...                          ! (other possible monitoring line records)
##### end control lines #####

##### control sections #####
! (this section is not active)
##### end control sections #####

##### section flow rate #####
! Input parameters on monitoring sections for the flow rate
n_sect dt_out n_fluid_types        ! n_sect(number of the flow rate monitoring
                             ! sections; dt_out(writing time step for
                             ! flow rates); n_fluid_types(number of fluid
                             ! types (the first ID fluid types are
                             ! selected)
first_section_ID                  ! first_section_ID=1
n_vertices                      ! n_vertices(number of vertices describing a
                             ! monitoring section for the flow rate (3 or
                             ! 4) of the first section
vertex_1_x vertex_1_y vertex_1_z
vertex_2_x vertex_2_y vertex_2_z
vertex_3_x vertex_3_y vertex_3_z
vertex_4_x vertex_4_y vertex_4_z
                             ! vertices of the first section (in case of
                             ! 3 vertices do not mind about the fourth
                             ! point)
...                          ! other possible section records
##### end section flow rate #####
```

Figure 11.1. Commented template of the main input file of SPHERA v.8.0.

11.3. Validation of SPHERA v.8.0

SPHERA v.8.0 is validated on 44 test cases (Figure 11.2). Some of them are published on International Journals and were also carried out with previous versions of the code. Other minor test cases only represent very simple configurations. The remaining 13 test cases are still matter of study for on-going numerical developments and validations, although preliminary validations are published.

Test case **Brief description or reference** **(indexed International Journal)**	**Example image**	**Test case** **Brief description or reference** **(indexed International Journal)**	**Example image**
"2D_erosional_dam_break_SPHERA_demo" Simple and demonstrative 2D erosional dam break (rough resolution) SPHERA v.8.0		"2jets_plate_DBSPH_high_res" Amicarelli et al. (2013, IJNME)+ "2jets_plate_DBSPH_low_res" Amicarelli et al. (2013, IJNME)+ "2jets_plate_SASPH_low_res" Amicarelli et al. (2013, IJNME)	
"Archimede" Simple and demonstrative test case: solid cube leaned on still water (rough resolution) SPHERA v.8.0		"asymmetric_wedge_20deg_light" Amicarelli et al. (2015, CAF)	
"asymmetric_wedge_20deg_medium" Amicarelli et al. (2015, CAF)		"Benchmark1_SASPH" Amicarelli et al. (2013, IJNME)+ "SPHERIC_B1_dam_break_DB-SPH" Amicarelli et al. (2013, IJNME)	
"Benchmark2_SASPH" Di Monaco et al. (2011, EACFM) + "SPHERIC_B2_dam_break_DB-SPH" Amicarelli et al. (2013, IJNME)		"body-body_impact_asymmetric" Amicarelli et al. (2015, CAF)+ "body-body_impact_low_vel" Amicarelli et al. (2015, CAF)+ "body-body_impact_symmetric" Amicarelli et al. (2015, CAF)	
"body-boundary_impact" Amicarelli et al. (2015, CAF)+ "body-boundary_impact_low_vel" Amicarelli et al. (2015, CAF)		"dam_break_2_bodies" Amicarelli et al. (2015, CAF)	
"dam_break_2D_demo" Simple and demonstrative test case on a 2D dam break (rough resolution) SPHERA v.8.0		"dam_break_body_UniBas" Amicarelli et al. (2015, CAF)	
"dam_break_multi-body" Amicarelli et al. (2015, CAF)		"erosional_dam_break_2D_FraCap02Taipei" Amicarelli & Agate (2014, SPHERIC)	
"erosional_dam_break_bed_2D_FraCap02" Amicarelli & Agate (2014, SPHERIC)		"erosional_dam_break_bed_2D_Spi05" Amicarelli & Agate (2014, SPHERIC)	

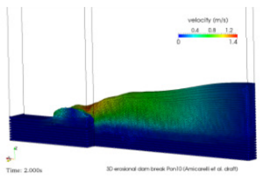
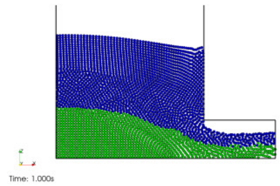
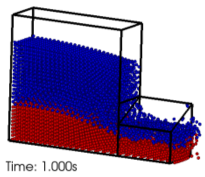
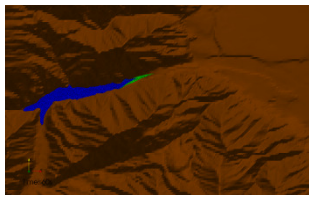
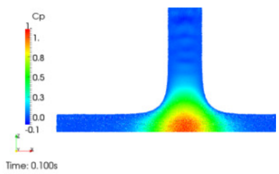
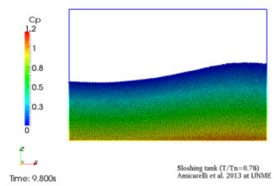
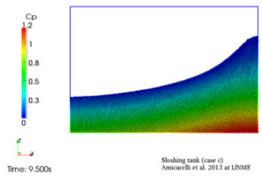
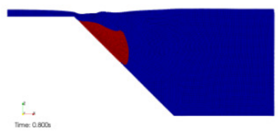
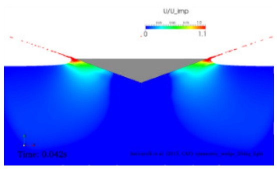
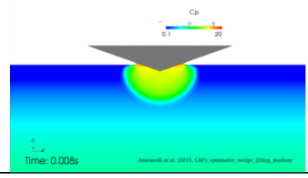
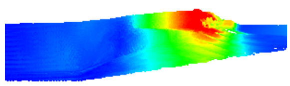
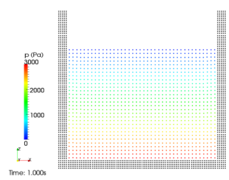
Test case **Brief description or reference** **(indexed International Journal)**	**Example image**	**Test case** **Brief description or reference** **(indexed International Journal)**	**Example image**
“erosional_dam_break_Pon10” Amicarelli & Agate (2014, SPHERIC)		“flushing_2D_small_granular_flows” Amicarelli & Agate (2014, SPHERIC)+ “flushing_2D_small_granular_flows_erosion_criterion_2D_complete” Amicarelli & Agate (2014, SPHERIC)+ “flushing_2D_small_tuned_model_and_erosion_criterion_2D_limited” (Manenti et al., 2012, JHE)	
“flushing_3D_small_slope_B3_granular_flows_erosion_criterion_2D_complete” SPHERA v.8.0 + “flushing_3D_small_slope_B3_granular_flows_erosion_criterion_3D_complete” SPHERA v.8.0		“ICOLD_earth-fill_dam_breach_long” Amicarelli & Agate (2014, SPHERIC) + “ICOLD_earth-fill_dam_breach_short” Amicarelli & Agate (2014, SPHERIC) + “ICOLD_earth-fill_dam_break” Amicarelli & Agate (2014, SPHERIC)	
“jet_plate_DBSPH“ Amicarelli et al. (2013, IJNME) + “jet_body-plate” Amicarelli et al. (2015, CAF)+ “jet_plate_DBSPH_low_res” Amicarelli et al. (2013, IJNME)+ “jet_plate_SASPH_low_res” Amicarelli et al. 2013 (IJNME)		“sloshing_tank_TbyTn_0_78” Amicarelli et al. (2013, IJNME)	
“sloshing_tank_TbyTn_1_07” Amicarelli et al. (2013, IJNME)		“submerged_landslide” Amicarelli & Agate (2014, SPHERIC)	
“symmetric_wedge_20deg_light” Amicarelli et al. (2015, CAF)		“symmetric_wedge_20deg_medium” Amicarelli et al. (2015, CAF)	
“tsunami_ISEC” Guandalini et al. (2014, DAOE)		“water_box_free_surface” Simple and demonstrative 2D test case on hydrostatic conditions (rough resolution) SPHERA v.8.0 + water_tank-body Simple and demonstrative 2D test case on hydrostatic conditions and "fluid - solid body" interactions (rough resolution) SPHERA v.8.0	

Figure 11.2. Validations and applications of SPHERA v.8.0 (44 test cases).
In azure those test cases, whose input files are published on SPHERA GitHub repository (Table 1.1).

12. FAQ

So far, no additional information is relevant.

13. BENEFICIARY LISTS ON THE LICENSES OF THE PREVIOUS VERSIONS OF THE CODE

SPHERA transfer on GitHub (as GNU-GPL FOSS) allows avoiding the production of ad-hoc licenses for limited periods. For sake of completeness, this section collects the exhaustive lists of SPHERA licenses (they are all expired), which has been realised by RSE SpA (formerly ERSE SpA, formerly CESI RICERCA SpA, formerly CESI SpA - Ricerca di Sistema) for the previous versions of the code (before SPHERA v.8.0).

Hereafter the list of beneficiaries of the licences (all expired) of SPHERA source codes:

- E4 Computer Engineering (license for company use; the license was exclusively released to translate the code in CUDA language, in collaboration with RSE SpA; 2009; SPHERA v.3.2);
- Prof. Mario Gallati (University of Pavia; personal license for single-user for academic use, in the frame of a collaboration with RSE SpA funded by Ricerca di Sistema; 2009; SPHERA v.3.2).

Hereafter the list of beneficiaries of the licences (all expired) of SPHERA executable files:

- Dr. Raffaele Albano and Dean Prof. Aurelia Sole (University of Basilicata; SPHERA v.3.3, 17May2012-16May2013; SPHERA v.4.0, 21May2012-20May2013; SPHERA v.5.0, 14Nov2012-13Nov2013; SPHERA v.5.0.2, 03Apr2013-31Oct2013; SPHERA v.5.0.2, 01Nov2013-31Dec2013; SPHERA v.5.0.2 –without erosion criterion -, 01Jul2015-31Dec15);
- Ms. Luciana Giuzio and Prof. Aurelia Sole (University of Basilicata; SPHERA v.3.3, 17May2012-16May2013);
- Prof. Philippe Larroudé (University of Grenoble -FRA-; SPHERA v.4.0.5 -without RK time integration schemes, without DB-SPH method-, 06Apr2012-05Apr2013);
- Ms. Latifa Ziane and Prof. Mohammed Chérif Khellaf (University USTHB of Algiers, Algeria; SPHERA v.7.0, 01Jul2015-31Dec2015).

14. PREVIOUS VERSIONS OF THE CODE

SPHERA v.8.0 includes all the relevant source files of the previous versions of the code.
Although these versions were not tracked on GitHub SPHERA repository (because the code was not written under a Git repository), the obsolete program units are kept as draft subroutines in (and only in) SPHERA v.8.0, just to keep them tracked on SPHERA GitHub repository. Further, Table 14.1 reports information on SPHERA v.7.0 files and their relationships with SPHERA v.8.0 program units.

Program unit (SPHERA v.7.0)	**First author (SPHERA v.7.0)**	**SPHERA 8.0 folder**	**SPHERA v.8.0 file**	**subroutine(s)/ function(f)/ module(m)/ main(p)**	**Notes**
CancelOutgoneParticles_2D	undef.	BC	CancelOutgoneParticles.f90	s	
CancelOutgoneParticles_3D	undef.	BC	CancelOutgoneParticles.f90	s	
FindFrame	undef.	BC	Sphera_Tools.f90	s	
FindLine	undef.	BC	Sphera_Tools.f90	s	
GenerateSourceParticles_2D	Di Monaco	BC	GenerateSourceParticles.f90	s	
GenerateSourceParticles_3D	Di Monaco	BC	GenerateSourceParticles.f90	s	
IsParticleInternal2D	undef.	BC	Sphera_Tools.f90	f	
IsParticleInternal3D	undef.	BC	Sphera_Tools.f90	f	
NormFix	undef.	BC	Sphera_Tools.f90	s	
NumberSectionPoints	undef.	BC	Sphera_Tools.f90	f	
PreSourceParticles_2D	Di Monaco	BC	GenerateSourceParticles.f90	s	
PreSourceParticles_3D	Di Monaco	BC	GenerateSourceParticles.f90	s	
VelLaw	undef.	BC	Sphera_Tools.f90	s	
CalcPre	undef.	BE_Mass	Sphera_Tools.f90	s	with commented subroutine on Mach check
inter_EqCont_2D	undef.	BE_Mass	Inter.f90	s	
inter_EqCont_3D	undef.	BE_Mass	Inter.f90	s	
inter_SmoothPres	Di Monaco	BE_Mass	Inter.f90	s	
PressureSmoothing_2D	Di Monaco	BE_Mass	PressureSmoothing.f90	s	
PressureSmoothing_3D	Di Monaco	BE_Mass	PressureSmoothing.f90	s	
diffumorris	undef.	BE_Momentum	Sphera_Tools.f90	s	
inter_EqMoto	undef.	BE_Momentum	Inter.f90	s	
inter_SmoothVelo_2D	Di Monaco	BE_Momentum	Inter.f90	s	
inter_SmoothVelo_3D	Di Monaco	BE_Momentum	Inter.f90	s	
viscomon	undef.	BE_Momentum	Sphera_Tools.f90	s	
viscomorris	undef.	BE_Momentum	Sphera_Tools.f90	s	
Body_dynamics_output	Amicarelli	Body_Transport	Body_dynamics.f90	s	
body_particles_to_continuity	Amicarelli	Body_Transport	Body_dynamics.f90	s	
body_pressure_mirror	Amicarelli	Body_Transport	Body_dynamics.f90	s	
body_pressure_postpro	Amicarelli	Body_Transport	Body_dynamics.f90	s	
body_to_smoothing_pres	Amicarelli	Body_Transport	Body_dynamics.f90	s	
body_to_smoothing_vel	Amicarelli	Body_Transport	Body_dynamics.f90	s	

Gamma_boun	Amicarelli	Body_Transport	Body_dynamics.f90	f	
Input_Body_Dynamics	Amicarelli	Body_Transport	Body_dynamics.f90	s	
RHS_body_dynamics	Amicarelli	Body_Transport	Body_dynamics.f90	s	
mixture_viscosity	Amicarelli	Constitutive_Equation	Granular_flows.f90	s	
viscapp	Di Monaco	Constitutive_Equation	Sphera_Tools.f90	s	
adjacent_faces_isolated_points	Amicarelli	DB_SPH	BC_wall_elements.f90	s	
BC_wall_elements	Amicarelli	DB_SPH	BC_wall_elements.f90	s	
DBSPH_find_close_faces	Amicarelli	DB_SPH	BC_wall_elements.f90	s	
DBSPH_IC_surface_elements	Amicarelli	DB_SPH	BC_wall_elements.f90	s	
DBSPH_inlet_outlet	Amicarelli	DB_SPH	BC_wall_elements.f90	s	
DBSPH_kinematics	Amicarelli	DB_SPH	BC_wall_elements.f90	s	
Gradients_to_MUSCL	Amicarelli	DB_SPH	BC_wall_elements.f90	s	
Gradients_to_MUSCL_boundary	Amicarelli	DB_SPH	BC_wall_elements.f90	s	
Import_ply_surface_meshes	Amicarelli	DB_SPH	BC_wall_elements.f90	s	
semi_particle_volumes	Amicarelli	DB_SPH	BC_wall_elements.f90	s	
viscomon_wall_elements	Amicarelli	DB_SPH	BC_wall_elements.f90	s	in drafts.f90
viscomorris_wall_elements	Amicarelli	DB_SPH	BC_wall_elements.f90	s	in drafts.f90
wall_elements_pp	Amicarelli	DB_SPH	BC_wall_elements.f90	s	
wavy_inlet	Amicarelli	DB_SPH	BC_wall_elements.f90	s	
compute_k_BetaGamma	Amicarelli	Erosion_Criterion	Granular_flows.f90	s	
fixed_bed_slope_limited	Amicarelli	Erosion_Criterion	Granular_flows.f90	s	
MohrC	Manenti	Erosion_Criterion	Crit_Erosion.f90	s	in drafts.f90
Shields	Manenti	Erosion_Criterion	Crit_Erosion.f90	s	
area_quadrilateral	Amicarelli	Geometry	Granular_flows.f90	s	
area_triangle	Amicarelli	Geometry	Granular_flows.f90	s	
dis_point_plane	Amicarelli	Geometry	Body_dynamics.f90	s	
distance_point_line_2D	Amicarelli	Geometry	Body_dynamics.f90	s	
distance_point_line_3D	Amicarelli	Geometry	Body_dynamics.f90	s	
IsPointInternal	undef.	Geometry	Sphera_Tools.f90	f	
line_plane_intersection	Amicarelli	Geometry	Granular_flows.f90	s	
LocalNormalCoordinates	Di Monaco	Geometry	Sphera_Tools.f90	s	
Matrix_Inversion_2x2	Amicarelli	Geometry	Body_dynamics.f90	s	
Matrix_Inversion_3x3	Amicarelli	Geometry	Body_dynamics.f90	s	
MatrixProduct	undef.	Geometry	Sphera_Tools.f90	s	
MatrixTransposition	undef.	Geometry	Sphera_Tools.f90	s	
point_inout_polygone	Amicarelli	Geometry	Body_dynamics.f90	s	
quadratic_equation	Amicarelli	Geometry	Granular_flows.f90	s	
reference_system_change	Amicarelli	Geometry	Body_dynamics.f90	s	
three_plane_intersection	Amicarelli	Geometry	Body_dynamics.f90	s	
Vector_Product	undef.	Geometry	Sphera_Tools.f90	s	
vector_rotation	Amicarelli	Geometry	Body_dynamics.f90	s	
GeneratePart	undef.	IC	Sphera_Tools.f90	s	
initialization_fixed_granular_particle	Amicarelli	IC	Granular_flows.f90	s	
SetParticleParameters	Amicarelli	IC	Sphera_Tools.f90	s	
SetParticles	undef.	IC	Sphera_Tools.f90	s	

SubCalcPreIdro	Agate	IC	Sphera_Tools.f90	s	
AggDens	undef.	Interface_Dispersion	Sphera_Tools.f90	s	in drafts.f90
inter_CoefDif	undef.	Interface_Dispersion	Inter.f90	s	in drafts.f90
inter_SmoothVF	Manenti	Interface_Dispersion	Inter.f90	s	in drafts.f90
check_files	undef.	Main algorithm	Sphera_Main.f90	s	in sphera.f90
Gest_Dealloc	undef.	Main algorithm	Sphera_Tools.f90	s	
Gest_Trans	undef.	Main algorithm	Sphera_Tools.f90	s	
Loop_Irre_2D	Di Monaco	Main algorithm	Loop_Irre.f90	s	
Loop_Irre_3D	undef.	Main algorithm	Loop_Irre.f90	s	
sphera	undef.	Main algorithm	Sphera_Main.f90	p	in sphera.f90
AdM_User_Type	Di Monaco	Modules	AdM_User_Type.f90	m	New name: Hybrid_allocation_module
ALLOC_Module	undef.	Modules	Alloc_Module.f90	m	New name: Dynamic_allocation_module
BoundIntegralTab_Module	Di Monaco	Modules	BoundIntegralTab_Module.f90	m	New name: SA_SPH_module
diagnostic_module	undef.	Modules	Diagnostic_Module.f90	m	New name: I_O_diagsnostic_module
english_writime2	undef.	modules	Sphera_Tools.f90	m	New name: I_O_ENG_module
files_entities	undef.	Modules	Files_Entities.f90	m	New name: I_O_file_module
GLOBAL_Module	undef.	Modules	Global_Module.f90	m	New name: Static_allocation_module
italiano_writime2	undef.	modules	Sphera_Tools.f90	m	New name: I_O_ITA_module
language_writime2	undef.	modules	Sphera_Tools.f90	m	New name: I_O_language_module
time_usertype	undef.	Modules	Time_UserType.f90	m	New name: Time_module
CalcVarLength	undef.	Neighbouring_Search	Sphera_Tools.f90	s	
CellIndices	undef.	Neighbouring_Search	Sphera_Tools.f90	f	
CellNumber	undef.	Neighbouring_Search	Sphera_Tools.f90	f	
CreaGrid	undef.	Neighbouring_Search	Sphera_Tools.f90	s	
InterFix	undef.	Neighbouring_Search	Inter.f90	s	
OrdGrid1	undef.	Neighbouring_Search	Sphera_Tools.f90	s	
ParticleCellNumber	undef.	Neighbouring_Search	Sphera_Tools.f90	f	
SearchforParticleZone_3D	Di Monaco	Neighbouring_Search	Sphera_Tools.f90	s	
w	Di Monaco	Neighbouring_Search	Boundaries.f90	f	
calc_pelo	undef.	Post-processing	Sphera_Tools.f90	s	
CalcVarp	undef.	Post-processing	Sphera_Tools.f90	s	
CreateSectionPoints	undef.	Post-processing	Sphera_Tools.f90	s	
GetVarPart	undef.	Post-processing	Sphera_Tools.f90	s	
Memo_Ctl	undef.	Post-processing	Sphera_Tools.f90	s	
Memo_Results	undef.	Post-processing	Sphera_Tools.f90	s	
Print_Results	undef.	Post-processing	Sphera_Tools.f90	s	
result_converter	undef.	Post-processing	Sphera_Tools.f90	s	
s_ctime	undef.	Post-processing	Sphera_Tools.f90	s	
s_secon2	undef.	Post-processing	Sphera_Tools.f90	s	

start_and_stop	Agate	Post-processing	Sphera_Tools.f90	s
sub_Q_sections	Amicarelli	Post-processing	Granular_flows.f90	s
Update_Zmax_at_grid_vert_columns	Amicarelli	Post-processing	Granular_flows.f90	s
write_Granular_flows_interfaces	Amicarelli	Post-processing	Granular_flows.f90	s
write_h_max	Amicarelli	Post-processing	Granular_flows.f90	s
writime2	undef.	Post-processing	Sphera_Tools.f90	s
defcolpartzero	undef.	Pre-processing	Sphera_Tools.f90	s
diagnostic	Agate	Pre-processing	Sphera_Tools.f90	s
Gest_Input	undef.	Pre-processing	Sphera_Tools.f90	s
Init_Arrays	undef.	Pre-processing	Sphera_Tools.f90	s
ModifyFaces	undef.	Pre-processing	Sphera_Tools.f90	s
ReadBedLoadTransport	Amicarelli	Pre-processing	ReadInputFile.f90	s
ReadBodyDynamics	Amicarelli	Pre-processing	ReadInputFile.f90	s
ReadCheck	undef.	Pre-processing	ReadInputFile.f90	f
ReadDBSPH	Amicarelli	Pre-processing	ReadInputFile.f90	s
ReadInput	undef.	Pre-processing	ReadInputFile.f90	s
ReadInputBoundaries	undef.	Pre-processing	ReadInputFile.f90	s
ReadInputControlLines	undef.	Pre-processing	ReadInputFile.f90	s
ReadInputControlPoints	undef.	Pre-processing	ReadInputFile.f90	s
ReadInputControlSections	undef.	Pre-processing	ReadInputFile.f90	s
ReadInputDomain	undef.	Pre-processing	ReadInputFile.f90	s
ReadInputDrawOptions	undef.	Pre-processing	ReadInputFile.f90	s
ReadInputExternalFile	undef.	Pre-processing	ReadInputFile.f90	s
ReadInputFaces	undef.	Pre-processing	ReadInputFile.f90	s
ReadInputGeneralPhysical	undef.	Pre-processing	ReadInputFile.f90	s
ReadInputLines	undef.	Pre-processing	ReadInputFile.f90	s
ReadInputMedium	undef.	Pre-processing	ReadInputFile.f90	s
ReadInputOutputRegulation	undef.	Pre-processing	ReadInputFile.f90	s
ReadInputParticlesData	undef.	Pre-processing	ReadInputFile.f90	s
ReadInputRestart	undef.	Pre-processing	ReadInputFile.f90	s
ReadInputRunParameters	undef.	Pre-processing	ReadInputFile.f90	s
ReadInputTitle	undef.	Pre-processing	ReadInputFile.f90	s
ReadInputVertices	undef.	Pre-processing	ReadInputFile.f90	s
ReadRestartFile	undef.	Pre-processing	ReadInputFile.f90	s
ReadRiga	undef.	Pre-processing	ReadInputFile.f90	s
ReadSectionFlowRate	Amicarelli	Pre-processing	ReadInputFile.f90	s
AddBoundaryContribution_to_CE2D	Di Monaco	SA_SPH	AddBoundaryContribution.f90	s
AddBoundaryContribution_to_CE3D	Di Monaco	SA_SPH	AddBoundaryContribution.f90	s
AddBoundaryContributions_to_	Di Monaco	SA_SPH	AddBoundaryContributi	s

ME2D			on.f90	
AddBoundaryContributions_to_ME3D	Di Monaco	SA_SPH	AddBoundaryContribution.f90	s
AddElasticBoundaryReaction_2D	Di Monaco	SA_SPH	AddBoundaryContribution.f90	s
AddElasticBoundaryReaction_3D	Di Monaco	SA_SPH	AddBoundaryContribution.f90	s
BoundaryMassForceMatrix2D	Di Monaco	SA_SPH	Boundaries.f90	s
BoundaryMassForceMatrix3D	Di Monaco	SA_SPH	Boundaries.f90	s
BoundaryPressureGradientMatrix3D	Di Monaco	SA_SPH	Boundaries.f90	s
BoundaryReflectionMatrix2D	Di Monaco	SA_SPH	Boundaries.f90	s
BoundaryVolumeIntegrals2D	Di Monaco	SA_SPH	Boundaries.f90	s
CompleteBoundaries3D	Di Monaco	SA_SPH	Boundaries.f90	s
ComputeBoundaryDataTab	Di Monaco	SA_SPH	Boundaries.f90	s
ComputeBoundaryIntegralTab	Di Monaco	SA_SPH	Boundaries.f90	s
ComputeBoundaryVolumeIntegrals_P0	Di Monaco	SA_SPH	Boundaries.f90	s
ComputeKernelTable	Di Monaco	SA_SPH	Boundaries.f90	s
ComputeSurfaceIntegral_WdS2D	Di Monaco	SA_SPH	Boundaries.f90	s
ComputeVolumeIntegral_WdV2D	Di Monaco	SA_SPH	Boundaries.f90	s
DefineBoundaryFaceGeometry3D	Di Monaco	SA_SPH	Boundaries.f90	s
DefineBoundarySideGeometry2D	Di Monaco	SA_SPH	Boundaries.f90	s
DefineBoundarySideRelativeAngles2D	Di Monaco	SA_SPH	Boundaries.f90	s
DefineLocalSystemVersors	Di Monaco	SA_SPH	Boundaries.f90	s
EvaluateBER_TimeStep	Di Monaco	SA_SPH	Boundaries.f90	s
FindBoundaryConvexEdges3D	Di Monaco	SA_SPH	Boundaries.f90	s
FindBoundaryIntersection2D	Di Monaco	SA_SPH	Boundaries.f90	s
FindCloseBoundaryFaces3D	DI Monaco	SA_SPH	Boundaries.f90	s
FindCloseBoundarySides2D	Di Monaco	SA_SPH	Boundaries.f90	s
GridCellBoundaryFacesIntersections3D	Di Monaco	SA_SPH	Boundaries.f90	s
InterpolateBoundaryIntegrals2D	Di Monaco	SA_SPH	Boundaries.f90	s
InterpolateTable	Di Monaco	SA_SPH	Boundaries.f90	s
IWro2dro	Di Monaco	SA_SPH	Boundaries.f90	f
J2Wro2	Di Monaco	SA_SPH	Boundaries.f90	f
JdWsRn	Di Monaco	SA_SPH	Boundaries.f90	f
SelectCloseBoundarySides2D	Di Monaco	SA_SPH	Boundaries.f90	s
WIntegr	Di Monaco	SA_SPH	Boundaries.f90	f
GetToken	undef.	strings	Sphera_Tools.f90	f
lcase	undef.	strings	Sphera_Tools.f90	f
ltrim	undef.	strings	Sphera_Tools.f90	f
Euler	Amicarelli	Time_Integration	time_integration.f90	s
Heun	Amicarelli	Time_Integration	time_integration.f90	s
inidt2	undef.	Time_Integration	Sphera_Tools.f90	s
rundt2	undef.	Time_Integration	Sphera_Tools.f90	s
stoptime	undef.	Time_Integration	Sphera_Tools.f90	s
time_integration	Amicarelli	Time_Integration	time_integration.f90	s
time_integration_body_dynamics	Amicarelli	Time_Integration	time_integration.f90	s

KeyDecoderCheck	Agate	KeyDecoderCheck.f90	s	Erased permanently
sloshing_tank_control_points	Amicarelli	DB-SPH_hard_coding.f90	s	Erased permanently

Table 14.1. File and Copyright information on SPHERA v.7.0; relationships with SPHERA v.8.0 program units.

15. SPHERA ACKNOWLEDGMENTS

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

- ✓ 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.

"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:

- ✓ HSPHMI14 - High performance computing for Lagrangian numerical models to simulate free surface and multi-phase flows (SPH) and the scalar transport in turbulent flows (MIcromixing); June 2014 – March 2015; Amicarelli A., G. Agate, G. Leuzzi, P. Monti, R. Guandalini, S. Sibilla; HPC Italian National Research Project (ISCRA-C2); competitive call for instrumental funds;
- ✓ HPCEFM15 - High Performance Computing for Environmental Fluid Mechanics 2015 (Italian National HPC Research Project); instrumental funding based on competitive calls (ISCRA-C project at CINECA, Italy); 2015 - in progress; Amicarelli A., A. Balzarini, S. Sibilla, G. Agate, G. Leuzzi, P. Monti, G. Pirovano, G.M. Riva, A. Toppetti, E. Persi, G. Petaccia, L. Ziane, M.C. Khellaf.

16. SPHERA REGISTRATION

SPHERA v.8.0 Copyright is registered (“Registro pubblico speciale per i programmi per elaboratore, SIAE”, Italy).

17. REFERENCES.

1. Adami S., X.Y. Hu, N.A. Adams; 2012; A generalized wall boundary condition for smoothed particle hydrodynamics; Journal of Computational Physics, 231:7057–7075.
2. Agate G., R. Guandalini; 2010; The Use of 3D SPHERA Code to Support Spillway Design and Safety Evaluation of Flood Events; Proceedings of the 5th International SPHERIC Workshop, pp.267-272. Edited by Ben Rogers, Copyright 5th International SPHERIC Workshop Local Organising Committee, Manchester (UK).
3. Amicarelli A., G. Agate; 2015; A 3D SPH integrated model for granular flows with transport of solid bodies: model couplings and enhanced boundary treatment based on surface wall elements; 10th SPHERIC ERCOFTAC International Workshop SPHERIC 2015 - ERCOFTACT Conference, 23-30; Parma (Italy).
4. Amicarelli A., G. Agate, 2014; SPH modelling of granular flows, 9th SPHERIC ERCOFTAC International ERCOFTAC Workshop:17-24, Paris
5. 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.
6. 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
7. Amicarelli A., J.-C. Marongiu, F. Leboeuf, J. Leduc, J. Caro; 2011; SPH truncation error in estimating a 3D function; Computers & Fluids, 44:279-296.
8. Amicarelli A., J.-C. Marongiu, F. Leboeuf, J. Leduc, M. Neuhauser, Le Fang, J. Caro; 2011; SPH truncation error in estimating a 3D derivative; International Journal for Numerical Methods in Engineering, 87(7):677-700.
9. Anghileri M., L.M.L. Castelletti, E. Francesconi, A. Milanese, M. Pittofrati; 2011; Rigid body water impact - experimental tests and numerical simulations using the SPH method; International Journal of Impact Engineering, 38:141-151.
10. Antoci C, Gallati M, Sibilla S.; 2007; Numerical simulation of fluid-structure interaction by SPH; Computers & Structures, 85(11-14):879-890.
11. Antuono M, Colagrossi A, Marrone S, Lugni C.; 2011; Propagation of gravity waves through a SPH scheme with numerical diffusive terms; Computer Physics Communications, 182(4):866-877.
12. Areti K., K. Hendrickson, D.K.P. Yue; 2013; SPH for incompressible free-surface flows. Part II: Performance of a modified SPH method; Computers & Fluids, 86(5): 510–536, DOI: 10.1016/j.compfluid.2013.07.016
13. Ataie-Ashtiani B., G. Shobeyri; 2008; Numerical simulation of landslide impulsive waves by incompressible smoothed particle hydrodynamics; Int. J. Numer. Meth. Fluids 56, 209–232.
14. Basa M., N.J. Quinlan, M. Lastiwka; 2009; Robustness and accuracy of SPH formulations for viscous flow; International Journal for Numerical Methods in Fluids, 60(10):1127-1148.
15. Bate M.R.; 2012; Stellar, brown dwarf and multiple star properties from a radiation hydrodynamical simulation of star cluster formation; Monthly Notices of the Royal Astronomical Society, 419(4):3115-3146.
16. Bauer A., V. Springel; 2012; Subsonic turbulence in smoothed particle hydrodynamics and moving-mesh simulations; Mon. Not. R. Astron. Soc., 423(3), 2558–2578, doi:10.1111/j.1365-2966.2012.21058.x
17. Berzi D., F.C. Bossi, E. Larcan; 2012; Collapse of granular-liquid mixtures over rigid, inclined beds; PHYSICAL REVIEW E 85, 051308:1-5.
18. Berzi D., J.T. Jenkins; 2008; Approximate analytical solutions in a model for highly concentrated granular-fluid flows; PHYSICAL REVIEW E 78, 011304:1-10.
19. Berzi D., J.T. Jenkins; 2011; Surface flows of inelastic spheres; PHYSICS OF FLUIDS 23, 013303:1-8.
20. Bonelli S., D. Marot; 2010; Micromechanical modeling of internal erosion; European Journal of Environmental and Civil Engineering, 15(8): 1207-1224.
21. Bonet J. and T.-S.L. Lok; “Variational and momentum preservation aspects of Smooth Particle Hydrodynamic formulations”; Comput. Methods Appl. Mech. Engrg., 180 (1999) 97-115.
22. Bonet J., M. X. Rodriguez-Paz; “Hamiltonian formulation of the variable-h SPH equations”; Journal of Computational Physics, 209 (2005), 541–558.
23. Bouscasse B., A. Colagrossi, S. Marrone, M. Antuono; 2013; Nonlinear water wave interaction with floating bodies in SPH; Journal of Fluids and Structures, 42: 112–129.
24. Bui Ha H., K. Sako, R. Fukagawa; 2007; Numerical simulation of soil–water interaction using smoothed particle hydrodynamics (SPH) method; Journal of Terramechanics, 44: 339–346
25. Bui Ha H., R. Fukagawa, K. Sako, S. Ohno; 2008; Lagrangian meshfree particles method (SPH) for large deformation and failure flows of geomaterial using elastic–plastic soil constitutive model; Int. J. Numer. Anal. Meth. Geomech., 32:1537–1570.
26. Busini V., E. Marzo, A. Callioni, R. Rota; 2011; Definition of a short-cut methodology for assessing earthquake-related Na-Tech risk; Journal of Hazardous Materials, 192(1); 329–339; DOI:10.1016/j.jhazmat.2011.05.022
27. Cao Z., Z. Yue, G. Pender; 2011; Landslide dam failure and flood hydraulics. Part I: experimental investigation; Nat Hazards, 59:1003–1019; DOI 10.1007/s11069-011-9814-8

28. Cao Z., Z. Yue, G. Pender; 2011; Landslide dam failure and flood hydraulics. Part II: coupled mathematical modelling; Nat Hazards; 59:1021–1045; DOI 10.1007/s11069-011-9815-7
29. Capone T., A. Panizzo, J.J. Monaghan; 2010; SPH modelling of water waves generated by submarine landslides; Journal of Hydraulic Research, 48, Extra Issue (2010), pp. 80–84.
30. Chauchat J., M. Médale; 2010; A three-dimensional numerical model for incompressible two-phase flow of a granular bed submitted to a laminar shearing flow; Comput. Methods Appl. Mech. Engrg. 199: 439–449.
31. Colagrossi A, Landrini M.; 2003; Numerical simulation of interfacial flows by smoothed particle Hydrodynamics; Journal of Computational Physics, 191(2):448-475.
32. Colombo P., F. Coleselli; 1996; Elementi di geotecnica:1-500; Zanichelli.
33. Costabile P.; 2006; Propagazione di onde di dam break su fondo erodibile; XXX Convegno di Idraulica e Costruzioni Idrauliche (IDRA).
34. De Leffe M., D. Le Touzé and B. Alessandrini; 2009; Normal Flux Method at the boundary for SPH; Proceedings of the 4th Spheric Workshop, 150-157, Nantes (France).
35. Delorme L., A.Colagrossi, A. Souto-Iglesias, R. Zamora-Rodriguez, E. Botia-Vera; 2009; A set of canonical problems in sloshing, Part I: Pressure field in forced roll-comparison between experimental results and SPH; Ocean Engineering, 36:168–178.
36. Dey S.; 1999; Sediment threshold; Applied Mathematical Modelling, 23: 399-417.
37. De Padova D., M. Mossa, S. Sibilla, E. Torti; 2013; 3D SPH modelling of hydraulic jump in a very large channel; Journal of Hydraulic Research, 51(2):158-173; DOI: 10.1080/00221686.2012.736883
38. Dilts G.A.; “Moving-least-squares-particle hydropdynamics – I. Consistency and stability”; International Journal for Numerical Methods in Engineering; (1999), 44, 1115-1155.
39. Di Monaco A., S. Manenti, M. Gallati, S. Sibilla, G. Agate, R. Guandalini; 2011; SPH modeling of solid boundaries through a semi-analytic approach. Engineering Applications of Computational Fluid Mechanics, 5(1):1–15.
40. Dubois F.; 2001; Partial Riemann problem, boundary conditions and gas dynamics; In Absorbing Boundaries and Layers, Domain Decomposition Methods: Applications to large scale computations, 16-77, Nova Science Publishers Inc.
41. Espa P., Sibilla S., Gallati M.; 2008; SPH simulations of a vertical 2-D liquid jet introduced from the bottom of a free-surface rectangular tank; Adv. Appl. Fluid Mech., 3:105-140.
42. Espa P. S. Sibilla; 2014; Experimental Study of the Scour Regimes Downstream of an Apron for Intermediate Tailwater Depth Conditions; Journal of Applied Fluid Mechanics, 7(4):611-624.
43. European Commission; 2003; Floods: European research for better predictions and management solutions; Press Release Database, Brussels, 13 October 2003, IP/03/1381; http://europa.eu/rapid/press-release_IP-03-1381_en.htm
44. Faltinsen O.M., O.F. Rognebakke, I.A. Lukovsky, A.N. Timokha; 2000; Multidimensional modal analysis of nonlinear sloshing in a rectangular tank with finite water depth; J Fluid Mech., 407, 201–34.
45. Fan X., C.X. Tang, C.J. van Westen, D. Alkema; 2012; Simulating dam-breach flood scenarios of the Tangjiashan landslide dam induced by the Wenchuan Earthquake; Nat. Hazards Earth Syst. Sci., 12:3031–3044.
46. Farahani, R., Dalrymple, R., Hérault, A., and Bilotta, G. (2014). ”Three-Dimensional SPH Modeling of a Bar/Rip Channel System.” J. Waterway, Port, Coastal, Ocean Eng., 140(1), 82–99.
47. Feldman J. and J. Bonet; 2007; Dynamic refinement and boundary contact forces in SPH with applications in fluid flow problems; Int. J. Numer. Meth. Engng; 72,295–324.
48. Ferrand M., D.R.P. Laurence, B.D. Rogers, D. Violeau; 2010; Improved time scheme integration approach for dealing with semi analytical boundary conditions in SPARTACUS2D; Proc. 5th International SPHERIC Workshop, 98-105.
49. Ferrand M., D.R. Laurence, B.D. Rogers, D. Violeau, C. Kassiotis; 2013; Unified semi-analytical wall boundary conditions for inviscid laminar or turbulent flows in the meshless SPH method: International Journal for Numerical Methods in Fluids, 71(4):446-472.
50. Ferrand M., D. Violeau, B.D. Rogers, D.R.P. Laurence; 2011; Consistent wall boundary treatment for laminar and turbulent flows in SPH; Proc. 6th International SPHERIC Workshop, 275-282.
51. Ferrari A., M. Dumbser, E. F. Toro, A. Armanini; 2009; A new 3D parallel SPH scheme for free surface flows; Computers & Fluids, 38:1203-1217.
52. Fraccarollo L., Capart H.; 2002; Riemann wave description of erosional dam-break flows; J. Fluid Mech., 461, 183-228.
53. Fraccarollo L., H. Capart, Y. Zech; 2003; A Godunov method for the computation of erosional shallow water transients; Int. J. Numer. Meth. Fluids, 41: 951–976.
54. Frenk C.S.; S.D.M. White, P. Bode, J.R. Bond, G.L. Bryan, R. Cen, H.M.P. Couchman, A.E. Evrard, N. Gnedin, A. Jenkins, A.M. Khokhlov, A. Klypin, J.F. Navarro, M.L. Norman, J.P. Ostriker, J.M. Owen, F.R. Pearce, U.-L. Pen, M. Steinmetz, P.A. Thomas, J.V. Villumsen, J.W. Wadsley, M.S. Warren, G. Xu, G. Yepes; 1999; The Santa Barbara Cluster Comparison Project: A Comparison of Cosmological Hydrodynamics Solutions; The Astrophysical Journal, 525(2):554-582.
55. Gallati M., G. Braschi, Simulazione lagrangiana di flussi con superficie libera in problemi di idraulica; 2000; in “L’ACQUA 5”.

56. GEO-SLOPE International Ltd; The Lower San Fernando Dam; QUAKE/W Example File: Lower SanFernando Dam.doc; www.geo-slope.com
57. Gómez-Gesteira M., R.A. Dalrymple; 2004; Using a 3D SPH Method for Wave Impact on a Tall Structure; J. Waterways, Port, Coastal, Ocean Engineering, 130(2):63-69.
58. Gotoh H., T. Sakai; 2006; Key issues in the particle method for computation of wave breaking; Coastal Engineering, 53:171–179.
59. Grabe, J., B. Stefanova; 2014; Numerical modeling of saturated soils, based on Smoothed Particle Hydrodynamics (SPH). Part 1: Seepage analysis. geotechnik 37(3):191-197.
60. Grenier N., M. Antuono, A. Colagrossi, D. Le Touzé, B. Alessandrini; 2009; An Hamiltonian interface SPH formulation for multi-fluid and free surface flows. Journal of Computational Physics, 228:8380–8393.
61. Groenenboom P.H.L., B.K. Cartwright; 2010; Hydrodynamics and fluid-structure interaction by coupled SPH-FE method; Journal of Hydraulic Research, 48(Issue SUPPL.1):61-73.
62. Gudehus G., R.O. Cudmani, A.B: Libreros-Bertini, M.M. Bühler; 2004; In-plane and anti-plane strong shaking of soil systems and structures; Soil Dynamics and Earthquake Engineering, 24(4):319-342; DOI: 10.1016/j.soildyn.2003.12.007
63. Guzzetti F.; 2015; Frane e alluvioni, una lunga storia italiana; ECOSCIENZA, 3:12-13.
64. Hashemi M.R., R. Fatehi, M.T. Manzari; 2012; A modified SPH method for simulating motion of rigid bodies in Newtonian fluid flows; International Journal of Non-Linear Mechanics, 47:626–638.
65. Hashemi M.R., R. Fatehi, M.T. Manzari; 2011; SPH simulation of interacting solid bodies suspended in a shear flow of an Oldroyd-B fluid; Journal of Non-Newtonian Fluid Mechanics, 166:1239–1252.
66. Hay A., A. Leroyer, M. Visonneau; 2006; H-adaptive Navier–Stokes simulations of free-surface flows around moving bodies; J Mar Sci Technol, 11:1–18.
67. Herle I., G. Gudehus; 1999; Determination of parameters of a hypoplastic constitutive model from properties of grain assemblies; Mechanics of Cohesive-Frictional Materials, 4(5):461-486.
68. ICOLD - Committee on Computational Aspects of Analysis and Design of Dams; 2013; 12th Benchmark Workshop on Numerical Analysis of Dams (Graz, Austria), Theme C.
69. Inutsuka S.; 2002; Reformulation of smoothed particle hydrodynamics with Riemann solver; Journal of Computational Physics, 179, 238–267.
70. Johnson G.R., S.R. Beissel; “Normalized Smoothing functions for impact computations”; Int. Jour. Num. Methods Eng. 1996, 39: 2725-2741.
71. Kajtar J.B.; J. J. Monaghan; 2010; On the dynamics of swimming linked bodies. European Journal of Mechanics B/Fluids, 29:377-386.
72. Kajtar J.B.; J.J. Monaghan; 2012; On the swimming of fish like bodies near free and fixed boundaries; European Journal of Mechanics B/Fluids, 33:1–13.
73. Kamrath P., Disse M., Hammer M., J. Kongeter; 2006; Assessment of Discharge through a Dike Breach and Simulation of Flood Wave Propagation; Natural Hazards, 38:63–78; DOI 10.1007/s11069-005-8600-x
74. Kirkpatrick, W.M.; 1965; Effects of Grain Size and Grading on the Shearing Behaviour of Granular Materials; Proceedings of the Sixth International Conference on Soil Mechanics and Foundation Engineering, 1:273-277.
75. Kleefsman K.M.T., G. Fekken, A.E.P. Veldman, B. Iwanowski, B. Buchner; 2005; A volume of-fluid based simulation method for wave impact problems; Journal of Computational Physics, 206, 363–393.
76. Komatina D., M. Jovanovic; 1997; Experimental study of steady and unsteady free surface flows with water-clay mixtures; Journal of Hydraulic Research, 35(5): 579-590.
77. Koukouvinis P.K., J.S. Anagnostopoulos, D.E. Papantonis; 2013; An improved MUSCL treatment for the SPH-ALE method: comparison with the standard SPH method for the jet impingement case; International Journal for Numerical Methods in Fluids, 71(9): 1152–1177. DOI: 10.1002/fld.3706
78. Kramer S.L.; 1996; Geotechnical Earthquake Engineering:1- 653; Prentice Hall.
79. Kumar D., A.K. Patra, E.B. Pitman, H. Chi; 2013; Parallel Godunov smoothed particle hydrodynamics (SPH) with improved treatment of Boundary Conditions and an application to granular flows; Computer Physics Communications, 184-10, 2277-2286.
80. Kvicinsky S., J.L. Kueny, F. Avellan, E. Parkinson; “Experimental and numerical analysis of free surface flows in a rotating bucket”; 1st IAHR Symposium on Hydraulic Machinery and Systems; 2002; 483-490.
81. Lajeunesse E, Mangeney-Castelnau a, Vilotte JP. Spreading of a granular mass on a horizontal plane. Physics of Fluids 2004; 16(7):2371, doi:10.1063/1.1736611.
82. Lambe T.W., R.V. Whitman; 1979; Soil mechanics:1-553; Wiley.
83. Landau L.D., E.M. Lifshitz; 1959; Fluid Mechanics (Volume 6 of A Course of Theoretical Physics); 1-551.
84. Lastiwka M., N. Quinlan, M. Basa; “Adaptive particle distribution for Smoothed Particle Hydrodynamics”; Int. J. Numer. Meth. Fluids, 47(2005), 1403–1409.
85. Le Touzé D., J. Biddiscombe, A. Colagrossi, E. Jacquin, F. Leboeuf, J.-C. Marongiu, N. Quinlan, A. Amicarelli, M. Antuono, D. Barcarolo, M. Basa, J. Caro, M. De Leffe, N. Grenier, P.-M. Guilcher, M. Kerhuel, Fang Le, L. Lobovský, S. Marrone, A. Marsh, G. Oger, E. Parkinson, J. Soumagne; 2011; Next-generation Multi-mechanics Simulation Engine in a Highly Interactive Environment; Procedia Computer Science 7, 292–293.

86. Leduc J., J.C. Marongiu, F. Leboeuf, M. Lance, E. Parkinson; 2010; Improvement of multiphase model using preconditioned Riemann solvers; Proceedings of the 5th International SPHERIC Workshop, 2010.
87. Leonardi M., T. Rung; 2013; SPH Modelling of Bed Erosion for Water/Soil-Interaction; 8th SPHERIC ERCOFTAC International Workshop, 139-144, Trondheim (Norway).
88. Le Touzé D., J. Biddiscombe, A. Colagrossi, E. Jacquin, F. Leboeuf, J.-C. Marongiu, N. Quinlan, A. Amicarelli, M. Antuono, D. Barcarolo, M. Basa, J. Caro, M. De Leffe, N. Grenier, P.-M. Guilcher, M. Kerhuel, Fang Le, L. Lobovský, S. Marrone, A. Marsh, G. Oger, E. Parkinson, J. Soumagne; 2011; Next-generation Multi-mechanics Simulation Engine in a Highly Interactive Environment; Procedia Computer Science 7, 292–293.
89. Lin P., Wu Y., Bai J., Lin Q.; 2011; A numerical study of dam-break flow and sediment transport from a quake lake; Journal of Earthquake and Tsunami, 5(5):401–428; DOI: 10.1142/S1793431111001169
90. Liu M.B., G.R. Liu; 2006; Restoring particle consistency in smoothed particle hydrodynamics; Applied Numerical Mathematics, 56:19–36.
91. Liu MB, Liu GR.; 2010; Smoothed Particle Hydrodynamics (SPH): an Overview and Recent Developments; Arch Comput Methods Eng, 17:25–76.
92. Liu M.B., G.R. Liu, K.Y. Lam; 2003; Constructing smoothing functions in smoothed particle hydrodynamics with applications; Journal of Computational and Applied Mathematics, 155:263-284.
93. Liu W.K., S. Jun, Y. F. Zhang; 1995; Reproducing Kernel Particle Methods; International Journal for Numerical Methods in Fluids, 20:1081-1106.
94. Liu X., H. Xu, S. Shao, P. Lin; 2013; An improved incompressible SPH model for simulation of wave-structure interaction Computers and Fluids 71, 113-123.
95. Lominé F., L. Scholtès, L. Sibille, P. Poullain ; 2013; Modeling of fluid–solid interaction in granular media with coupled lattice Boltzmann/discrete element methods: application to piping erosion; Int. J. Numer. Anal. Meth. Geomech, 37:577–596.
96. Lube G, Huppert HE, Sparks RSJ, Hallworth Ma. Axisymmetric collapses of granular columns. Journal of Fluid Mechanics, Jun 2004; 508:175–199, doi:10.1017/S0022112004009036.
97. Macia F., L.M. Gonzalez, J.L. Cercos-Pita; A. Souto-Iglesias; 2012; A Boundary Integral SPH Formulation - Consistency and Applications to ISPH and WCSPH-; Progress of Theoretical Physics, 128-3, 439-462.
98. 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.
99. Marongiu J.-C.; 2007; Méthode numérique lagrangienne pour la simulation d'écoulements à surface libre - Application aux turbines Pelton; PhD thesis, Ecole Centrale de Lyon (France).
100. Marongiu J.C.; F. Leboeuf, J. Caro, E. Parkinson; 2010; Free surface flows simulations in Pelton turbines using an hybrid SPH-ALE method; J. Hydraul. Res., 47:40–49.
101. Marongiu J.C., F. Leboeuf, E. Parkinson; 2007; Numerical simulation of the flow in a Pelton turbine using the meshless method smoothed particle hydrodynamics: a new simple solid boundary treatment; Proceeding of the institution of mechanical engineering- Part A, Journal of Power and Energy, 221-A6, 849-856.
102. Marrone S., B. Bouscasse, A. Colagrossi, M. Antuono; 2012; Study of ship wave breaking patterns using 3D parallel SPH simulations; Computers & Fluids, 69:54–66.
103. Marrone S, Colagrossi A, Antuono M, Lugni C, Tulin M.P.: 2011; A 2D+t SPH model to study the breaking wave pattern generated by fast ships; Journal of Fluids and Structures, 27:1199-1215.
104. Maurel B., S. Potapov, J. Fabis, A. Combescure; 2009; Full SPH fluid–shell interaction for leakage simulation in explicit dynamics ; International Journal for Numerical Methods in Engineering, 80(2):210–234.
105. Mayrhofer A., B.D. Rogers, D. Violeau, M. Ferrand; 2013; Investigation of wall bounded flows using SPH and the uni fied semi-analytical wall boundary conditions; Computer Physics Communications, 184: 2515–2527.
106. Mayrhofer A, Rogers BD, Violeau D, Ferrand M.; 2013; Investigation of wall bounded flows using SPH and the uni fied semi-analytical wall boundary conditions. Computer Physics Communications 184, 2515–2527.
107. Mirauda D., M. Greco, A. Volpe Plantamura; 2011; Influence of the entropic parameter on the flow geometry and morphology; Proc. World Academy of Science, Engineering and Technology, ICWEE 2011, Phuket, Thailand; 1357–1362.
108. Molteni D., A. Colagrossi; 2009; A simple procedure to improve the pressure evaluation in hydrodynamic context using the SPH; Computer Physics Communications, 180:861–872.
109. Monaghan J. J.; 1992; Smoothed Particle Hydrodynamics; Annu. Rev. Astron. Astrophys, 30, 543-74.
110. Monaghan J.J. 1994; Simulating Free Surface Flows with SPH; Journal of Computational Physics, 110, 399-406.
111. Monaghan JJ. Smoothed particle hydrodynamics; 2005; Rep. Prog. Phys., 68:1703–1759.
112. Monaghan J.J., J.C. Lattanzio; 1985; A refined method for astrophysical problems. Astronomy and Astrophysics, 149, 135-143.
113. Monaghan J.J., J.B. Kajtar; 2009; SPH particle boundary forces for arbitrary boundaries. Computer Physics Communications, 180:1811–1820.
114. Monaghan J.J., A. Kos, N. Issa; 2003; Fluid motion generated by impact; Journal of Waterway, Port, Coastal and Ocean Engineering, 129:250–259.
115. Monaghan J.J., A. Raflee; 2013; A simple SPH algorithm for multi-fluid flow with high density ratios; International Journal for Numerical Methods in Fluids, 71(5):537-561.

116. Morrison F. A.; 2013; An Introduction to Fluid Mechanics, Cambridge University Press, New York.
117. Moulinec C., R.Issa, J.C. Marongiu, D. Violeau; "Parallel 3-D SPH simulations"; CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES, 25-3 (2008) , 133-147.
118. Nestor R.M., N.J. Quinlan; 2013; Application of the meshless finite volume particle method to flow-induced motion of a rigid body; Computers & Fluids, 88:386–399.
119. Neuhauser M., J.-C. Marongiu, F. Leboeuf; 2013; Coupling of the meshless SPH-ALE method with a finite volume method; Particle-based Methods III: Fundamentals and Applications – Proceedings of the 3rd International Conference on Particle based Methods, Fundamentals and Applications, Particles 2013: 939-948.
120. Nguyen H.H. , D. Marot, F. Bendahmane; 2012; Erodibility characterisation for suffusion process in cohesive soil by two types of hydraulic loading; Houille Blanche, 6: 54-60.
121. Niemunis, A., Herle I.; 1997; Hypoplastic model for cohesionless soils with elastic strain range, Mechanics of Cohesive-Frictional Materials, 2 (4), pp. 279-299.
122. Omidvar P., P.K. Stansby, B.D. Rogers; 2012; SPH for 3D floating bodies using variable mass particle distribution; International Journal for Numerical Methods in Fluids, 72(4):427-452; DOI: 10.1002/fld.3749
123. Omidvar P., P.K. Stansby, B.D. Rogers; 2012; Wave body interaction in 2D using smoothed particle hydrodynamics (SPH) with variable particle mass; International Journal for Numerical Methods in Fluids, 68:686–705.
124. Oger G., M. Doring, B. Alessandrini, P. Ferrant; 2006; Impulse-based rigid body interaction in SPH; Journal of Computational Physics, 213:803–822.
125. Oger G., M. Doring, B. Alessandrini and P. Ferrant; 2007; "An improved SPH method: Towards higher order convergence"; Journal of Computational Physics 225:1472–1492.
126. Pasqualini; 1983; Standard penetration test S.P.T.; Conferenze di Geotecnica di Torino, XI ciclo, "Parametri di progetto da prove in situ": 1-94; 28Nov1983-01Dec1983, Torino (Italy).
127. Patel M.H., R. Vignjevic, J.C. Campbell; 2009; An SPH Technique for Evaluating the Behaviour of Ships In Extreme Ocean Waves; International Journal of Maritime Engineering, 151:39-47.
128. Pastor M., B. Haddad, G. Sorbino, S. Cuomo, V. Drempetic; 2009; A depth-integrated, coupled SPH model for flow-like landslides and related phenomena; Int. J. Numer. Anal. Meth. Geomech., 33:143–172.
129. Patel M.H., R. Vignjevic, J.C. Campbell; 2009; An SPH Technique for Evaluating the Behaviour of Ships In Extreme Ocean Waves; International Journal of Maritime Engineering, 151, 39-47.
130. Peralta C., A. Melatos, M. Giacobello, A. Ooi; 2008; Superfluid spherical Couette flow; Journal of Fluid Mechanics, 609:221-274; DOI: 10.1017/S002211200800236X
131. Pontillo M.; 2010; Trasporto ed "entrainment" di sedimenti in alvei mobili; PhD thesis, Università degli studi di Napoli Federico II.
132. Price, D.J.; 2012; Smoothed Particle Hydrodynamics and Magnetohydrodynamics; J. Comp. Phys., 231(3): 759-794.
133. Price D.J.; C. Federrath; 2010; A comparison between grid and particle methods on the statistics of driven, supersonic, isothermal turbulence; Monthly Notices of the Royal Astronomical Society, 406(3):1659-1674.
134. Price, D.J., J.J. Monaghan; 2007; An energy conserving formalism for adaptive gravitational force softening in SPH and N-body codes; MNRAS, 374, 1347-1358.
135. Progetto AVI - Aree Vulnerate Italiane; CNR; 1999-2016; http://sici.irpi.cnr.it/avi.htm
136. Quinlan N. J., M. Basa, M. Lastiwka; "Truncation error in mesh-free particle methods"; Int. J. Numer. Meth. Engng, 66 (2006), 2064–2085.
137. Randles R. W., L.D. Libertsky; 1996; Smoothed Particle Hydrodynamics. Some recent improvements and applications; Comput. Methods Appl. Mech. Engrg., 139:375-408.
138. Regazzoni P.-L., D. Marot; 2013; A comparative analysis of interface erosion tests; Nat Hazards, 67:937–950.
139. DOI 10.1007/s11069-013-0620-3
140. Rzadkiewicz, S., C. Mariotti, and P. Heinrich (1997). Numerical simulation of submarine landslides and their hydraulic effects. Journal of Waterway, Port, Coastal and Ocean Eng. 123 (4), 149-157.
141. Salvati P., C. Bianchi, M. Rossi, F. Guzzetti; 2010; Societal landslide and flood risk in Italy; Nat. Hazards Earth Syst. Sci., 10:465-483, doi:10.5194/nhess-10-465-2010
142. Schmocker L., W.H. Hager; 2012; Plane dike-breach due to overtopping: effects of sediment, dike height and discharge; Journal of Hydraulic Research, 50(6): 576-586.
143. Seed H.B., Idriss I.M., Lee K.L., Makdisi F.I.; 1975; Dynamic analysis of the slide in the lower san Fernando dam during the earthquake of February 9, 1971; ASCE J Geotech Eng Div, 101(9):889-911.
144. Seminara, G., L. Solari, G. Parker; 2002; Bed load at low Shields stress on arbitrarily sloping beds: Failure of the Bagnold hypothesis: Water Resour. Res. 38(11): 1249, doi:10.1029/2001WR000681.
145. Seungtaik O., K. Younghee, R. Byung-Seok; 2009; Impulse-based Boundary Force (IBF). Computer, animation and virtual worlds, 20:215–224.
146. Shao J.R., H.Q. Li, G.R. Liu, M.B. Liu; 2012; An improved SPH method for modeling liquid sloshing dynamics; Computers and Structures, 100-101:18–26.
147. Shepard D.; 1968; A two-dimensional interpolation function for irregularly spaced points. Proceedings of A.C.M National Conference, 517–524, New York (USA).

148. Shields A.; 1936; Application of similarity principles and turbulence research to bed-load movement; Ph.D. dissertation, Institute of Technology, Berlin (in German).
149. Shiels D., A. Leonard, A. Roshko; 2001; Flow-induced vibration of a circular cylinder at limiting structural parameters, Journal of Fluids and Structures, 15(1):3–21, doi:10.1006/jfls.2000.0330
150. Singh V.P., P.D. Scarlatos, J.G. Collins, M.R. Jourdan; 1988; Breach erosion of earthfill dams (BEED) model; Natural Hazards, 1(2):161-180.
151. “SPHERA v.8.0” (RSE SpA), https://github.com/AndreaAmicarelliRSE/SPHERA
152. Spinewine B.; 2005; Two-layer flow behavior and the effects of granular dilatancy in dam-break induced sheet-flow; PhD Thesis, Faculté des sciences appliquées, Université catholique de Louvain.
153. Soares-Frazão S.A., Y. Zech; 2007; Experimental study of dam-break flow against an isolated obstacle; Journal of Hydraulic Research, 45(Supplement 1):27-36.
154. Soares-Frazao S., Y. Zech; 2011; HLLC scheme with novel wave-speed estimators appropriate for two-dimensional shallow-water flow on erodible bed; Int. J. Numer. Meth. Fluids, 66:1019–1036.
155. Souto-Iglesias A., L. Delorme, L. Pérez-Rojas, S. Abril-Pérez; 2006; Liquid moment amplitude assessment in sloshing type problems with smooth particle hydrodynamics; Ocean Engineering, 33, 1462–1484.
156. Souto-Iglesias A., F. Macià, L.M. González, J.L. Cercos-Pita; 2013; On the consistency of MPS; Computer Physics Communications, 184-3, 732-745.
157. Souto Iglesias A., L. Perez Rojas, R.Z. Rodriguez; 2004 ; Simulation of anti-roll tanks and sloshing type problems with smoothed particle hydrodynamics; Ocean Engineering, 31:1169–1192.
158. Spinewine B.; 2005; Two-layer flow behavior and the effects of granular dilatancy in dam-break induced sheet-flow; PhD Thesis, Faculté des sciences appliquées, Université catholique de Louvain.
159. Stefanova, B., J. Grabe; 2014; SPH model of water jet erosion in granular soils with a boundary layer of liquefied soil; International Symposium on Geomechanics Micro to Macro, September 1-3, 2014, Cambridge, UK.
160. Stefanova B., K. Seitz, J. Bubel, J. Grabe; 2012; Water-Soil Interaction Simulation using Smoothed Particle Hydrodynamics; ICSE6 Paris (27-31 August 2012), ICSE6-201:695-704.
161. Swartenbroekx C., Y. Zech, S. Soares-Frazão; 2013; Two-dimensional two-layer shallow water model for dam break flows with significant bed load transport; Int. J. Numer. Meth. Fluids, 73(5):477–508; DOI: 10.1002/fld.3809
162. Szewc K., J. Pozorski, J.-P. Minier; 2012; Analysis of the incompressibility constraint in the smoothed particle hydrodynamics method; International Journal for Numerical Methods in Engineering, 92(4):343–369, DOI: 10.1002/nme.4339
163. Terzaghi, K.; 1943; Theoretical soil mechanics; New York, London: Wiley.
164. Torti E., S. Sibilla, M. Raboni; 2013; An Eulerian–Lagrangian method for the simulation of the oxygen concentration dissolved by a two-phase turbulent jet system; Computers & Structures, 129:207–217.
165. Ulrich, C.; 2013; Smoothed-Particle-Hydrodynamics simulation of port hydrodynamics problems; Schirftenreihe Schiffbau der Technischen Universität Hamburg-Harburg, Report Nr. 671. PhD Thesis. Technische Universität Hamburg-Harburg, Hamburg. Schiffbau.
166. USGS; 2012; The Los Angeles dam story; http://earthquake.usgs.gov/learn/publications/la-damstory/
167. Vacondio R, Rogers BD, Stansby PK, Mignosa P, Feldman J.; 2013; Variable resolution for SPH: a dynamic particle coalescing and splitting scheme; Comput. Methods Appl. Mech. Engrg., 256:132-148, doi: http://dx.doi.org/10.1016/j.cma.2012.12.014
168. Vacondio R, Rogers BD, Stansby P, Mignosa P.; 2012; SPH Modeling of Shallow Flow with Open Boundaries for Practical Flood Simulation; J. Hydraul. Eng., 138(6):530–541.
169. Valizadeh A., J.J. Monaghan; 2012; Smoothed particle hydrodynamics simulations of turbulence in fixed and rotating boxes in two dimensions with no-slip boundaries; Physics of Fluids, 24(3), Article number 035107.
170. Van Leer B.; “Towards the ultimate conservative difference scheme v. a second order sequel to Godunov’s method”; Journal of Computational Physics; 32 (1979), 101-136.
171. Van Rijn L.C.; 1982; The prediction of Bed-Forms and Alluvial Roughness; Report, Delft Hydraulics laboratory; The Netherlands.
172. Van Rijn L.C.; 1993; Principles of sediment transport in rivers, estuaries, and coastal seas; Aqua Publications.
173. Vaughan G.L.; 2009; The SPH equations for fluids. International Journal for Numerical Methods in Engineering, 79:1392–1418.
174. Vila J.P.; 1999; On particle weighted methods and Smooth Particle Hydrodynamics; Mathematical Models and Methods in Applied Sciences, 9(2):161-209.
175. Violeau D.; 2012; Fluid mechanics and the SPH method: theory and applications. Oxford University Press, Oxford.
176. Violeau D.; A. Leroy; 2014; On the maximum time step in weakly compressible SPH; Journal of Computational Physics, 256: 388-415.
177. von Wolffersdorff, P.A.; 1996; A hypoplastic relation for granular materials with a predefined limit state surface; Mechanics of Cohesive-Frictional Materials, 1(1):251–271.
178. Xu L., A.W. Troesch, R. Petterson; 1998; Asymmetric hydrodynamic impact and dynamic response of vessels; Proceedings of 17th International Conference on Offshore Mechanics and Arctic Engineering, 98-320.

179. Wang X., L.B. Wang; 2007; Dynamic analysis of a water-soil-pore water coupling system; Computers and Structures, 85(11-14):1020-1031; DOI: 10.1016/j.compstruc.2006.11.017
180. Wendland H., "Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree", Adv. Comput. Math. 1995, 4: 389-396.
181. Wiberg P.L., J.D. Smith; 1987; Calculations of the critical shear stress for motion of uniform and heterogeneous sediments; Water Resour. Res., 23(8):1471-1480.
182. Zoppé B.; 2004; Simulation numérique et analyse de l'écoulement dans les augets des turbines Pelton; PhD thesis; Institut national polytechnique (Grenoble, France).

18. GNU FREE DOCUMENTATION LICENSE

GNU Free Documentation License
Version 1.3, 3 November 2008

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
D. Preserve all the copyright notices of the Document.
E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
F. Include, immediately after the copyright notices, a license notice

giving the public permission to use the Modified Version under the
terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

    Copyright (c) YEAR YOUR NAME.
    Permission is granted to copy, distribute and/or modify this document
    under the terms of the GNU Free Documentation License, Version 1.3
    or any later version published by the Free Software Foundation;
    with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
    A copy of the license is included in the section entitled "GNU
    Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

    with the Invariant Sections being LIST THEIR TITLES, with the
    Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.