

SPHERA v.9.0 (RSE SpA): documentation

1. DESCRIPTION AND REFERENCES	3
2. WARRANTIES AND RESPONSABILITIES	5
3. CITATION OF SPHERA V.9.0.0	6
4. SPHERA DEVELOPERS/AUTHORS	7
5. SPHERA OFFICIAL USERS	8
6. THE SCHEME FOR TRANSPORT OF SOLID BODIES AND THE SEMI-ANALYTIC APPROACH AS A BOUNDARY TREATMENT SCHEME FOR FIXED BOUNDARIES	15
6.1. Smoothed Particle Hydrodynamics (SPH)	15
6.2. SPH approximation of the balance equations of fluid dynamics with the boundary treatment scheme of Di Monaco et al. (2011; semi-analytic approach)	16
6.3. SPH balance equations for rigid body transport	18
6.4. Improving 3D rotations	19
6.5. Sliding friction force	21
6.5.1. Aerial stage (non-negative value for the input friction angle and body-frontier interactions)	21
6.5.2. Aerial stage (negative value for the input friction angle or body-body interactions)	22
6.5.3. Submerged stage	23
6.6. Body-boundary normal reaction force under sliding (at null normal velocity)	23
6.6.1. Aerial stage	23
6.6.2. Submerged stage	23
6.7. Fluid-body interaction terms	23
6.8. Solid-solid interaction terms	25
6.9. Normal restitution coefficient	27
7. THE SCHEME FOR DENSE GRANULAR FLOWS	28
7.1. Mixture model for dense granular flows	28
7.2. 2-interface 3D erosion criterion	33
7.3. A simplified approach for soil liquefaction	36
8. THE DB-SPH BOUNDARY TREATMENT SCHEME	37
8.1. DB-SPH particle approximation and modifications of the balance equations	37
8.2. 1D Linearized Partial Riemann Solver	39
8.3. Semi-particle volume	39
8.4. DB-SPH inlet and outlet sections	40
8.5. Shear stress boundary terms	40
9. TIME INTEGRATION SCHEMES (LEAPFROG, EULER, HEUN)	42
10. THE SUBSTATION-FLOODING DAMAGE SCHEME	44
10.1. Mathematical models	44
10.1.1. Proxy damage and vulnerability to flood-induced blackout events, in the absence of redundancy (at the level of the single electrical substations)	44
10.1.2. Flood-induced damage to the components of the electrical substations	45
10.2. The numerical models	46
11. THE MODELLING CHAIN	47
12. DEVELOPER GUIDE	50
12.1. SPHERA v.9.0.0: synthetic description of the program units	50
12.1.1. Program units for the boundary conditions ("BC")	50
12.1.2. Program units for the continuity equation	50
12.1.3. Program units for the momentum equation	50
12.1.4. Program units for the transport of solid bodies	50
12.1.5. Program units for the constitutive equation	50
12.1.6. Program units for the boundary treatment scheme DB-SPH	51
12.1.7. Program units for the erosion criterion	52
12.1.8. Program units on geometry (i.e., analytic geometry, algebra, ...)	52
12.1.9. Program units for the initial conditions (IC)	54
12.1.10. Draft program units for the turbulent dispersion of granular material	54
12.1.11. Program units for the main algorithms	54
12.1.12. Modules	55
12.1.13. Program units for the neighbouring search, the smoothing operators and the interface detection.	55
12.1.14. Program units for post-processing	55
12.1.15. Program units for pre-processing	55
12.1.16. Program units for the boundary treatment scheme SA-SPH	60

12.1.18.	Program units for time integration	60
12.2.	Style formatting	60
12.3.	Modifications with respect to SPHERA v.8.0	61
13.	USER GUIDE	84
13.1.	Installation	84
13.2.	Commented template of the main input file of SPHERA v.9.0.0	84
13.3.	Tutorials	98
13.3.1.	“body_body_impacts”	98
13.3.2.	“body_boundary_impacts”	98
13.3.3.	“dam_breach_ICOLD_trunks”	98
13.3.4.	“db_2bodies”	98
13.3.5.	“db_2D_demo”	98
13.3.6.	“db_Alpe_Gera”	98
13.3.7.	“db_Alpe_Gera_Lanzada_substations”	98
13.3.8.	“db_body_exp_UniBas”	98
13.3.9.	“db_ICOLD”	98
13.3.10.	“db_multi_body”	98
13.3.11.	“db_squat_obstacle”	98
13.3.12.	“db_tall_obstacle”	99
13.3.13.	“dike_breach_2D_expSchHag12JHR_ID22”	99
13.3.14.	“edb_2D_demo”	99
13.3.15.	“edb_2D_FraCap02”	99
13.3.16.	“edb_2D_FraCap02_Taipei”	99
13.3.17.	“edb_2D_Spi05”	99
13.3.18.	“edb_ICOLD”	99
13.3.19.	“edb_KarlSand”	99
13.3.20.	“edb_Pon10”	99
13.3.21.	“floating_cube_stability”	99
13.3.22.	“flushing_2D”	99
13.3.23.	“flushing_3D”	99
13.3.24.	“jet_plate”	99
13.3.25.	“rectangular_side_weir_Fr_0_491”	99
13.3.26.	“San_Fernando_Lower_van_Norman_dam_liquefaction”	100
13.3.27.	“sloshing_tank_TbyTn_0_78”	100
13.3.28.	“sloshing_tank_TbyTn_1_07”	100
13.3.29.	“spherical_Couette_flows”	100
13.3.30.	“SPH_udb_exp_Kim2015HYDROL”	100
13.3.31.	“submerged_landslide”	100
13.3.32.	“still_water_tank”	100
13.3.33.	“wave_motion_for_WaveSAX”	100
13.3.34.	“wedges_falls_on_still_water”	100
14.	SPHERA ACKNOWLEDGMENTS	101
15.	SPHERA REGISTRATION	103
16.	REFERENCES	104
17.	GNU FREE DOCUMENTATION LICENSE	112

Documentation Copyright:

Copyright (C) 2015-2018 RSE SpA. 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 of this document is included in the section entitled "GNU Free Documentation License". This documentation represents an improvement of the previous version of the document (“SPHERA v.8.0 (RSE SpA): documentation”). The email address of the author of this documentation is: andrea.amicarelli@rse-web.it .

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.9.0.0 (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 (with transport of solid bodies and bed-load transport flood-control works, flood-induced damage; domain spatial coverage of some hundredths of squared kilometres) and fast landslides, sloshing tanks, sea waves and sediment removal from water reservoirs.

With Copyright 2005-2018 (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, Qiao Cheng.

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

- Scheme for dense granular flows. Reference: Amicarelli et al. (2017, IJCFD, [9]):
Amicarelli A., B. Kocak, S. Sibilla, J. Grabe; 2017; A 3D Smoothed Particle Hydrodynamics model for erosional dam-break floods; International Journal of Computational Fluid Dynamics, 31(10):413-434; DOI 10.1080/10618562.2017.1422731
- 3D SPH numerical scheme for the transport of solid bodies in free surface flows. Reference: Amicarelli et al. (2015, CAF, [7]):
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, [6]):
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, [122]):
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, [46]):
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 (e.g., 3D erosion criterion also with mixture-fixed bed interactions), 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 optimized executions
debug	SPHERA executable files compiled with gfortran/ifort for debug scalar executions
debug_omp	SPHERA executable files compiled with gfortran/ifort for debug parallel 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 v.9.0.0 repository.

2. WARRANTIES AND RESPONSABILITIES

SPHERA v.9.0.0 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.9.0.0

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.9.0.0 need the following citation: “SPHERA v.9.0.0 (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).

SPHERA should also be cited 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.9.0.0 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.9.0.0 is available on GitHub ([184]). 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 citations);
- 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);
- 7) be interested in attending free internships on SPHERA at RSE as academic students (Bachelor Degree, Master Degree and PhD students).

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 (RSE can delete program units since a certain version).

SPHERA authors and “official users” need to activate a free account on github.com (with recognizable name, surname and affiliation) and “fork” SPHERA, by clicking on the icon “fork” in the official SPHERA public repository ([184; 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).

AndreaAmicarelliRSE / SPHERA Watch 9 Star 10 Fork 14

[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Insights](#)

Tree: 55981ad8da **SPHERA / bin /** Create new file Find file History

AndreaAmicarelliRSE and Andrea Amicarelli SPHERA free software (FOSS). Minor modifications.		Latest commit c494a9f on 12 Nov 2015
..		
SPHERA_v_8_0_FOSS_gfortran_debug.x	SPHERA free software (FOSS). Minor modifications.	3 years ago
SPHERA_v_8_0_FOSS_gfortran_run.x	SPHERA free software (FOSS). Minor modifications.	3 years ago
SPHERA_v_8_0_FOSS_ifort_debug.x	SPHERA free software (FOSS). Minor modifications.	3 years ago
SPHERA_v_8_0_FOSS_ifort_run.x	SPHERA free software (FOSS). Minor modifications.	3 years ago


© 2018 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)  [Contact GitHub](#) [API](#) [Training](#) [Shop](#) [Blog](#) [About](#)

Figure 5.1. SPHERA on GitHub: first FOSS release (SPHERA v.8.0). Executable codes.

AndreaAmicarelliRSE / SPHERA Watch 9 Star 10 Fork 14

[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Insights](#)

Tree: 55981ad8da **SPHERA / doc /** Create new file Find file History

AndreaAmicarelliRSE and Andrea Amicarelli SPHERA free software (FOSS). Minor modifications.		Latest commit c494a9f on 12 Nov 2015
..		
SPHERA_v_8_0_documentation_FOSS.pdf	SPHERA free software (FOSS). Minor modifications.	3 years ago


© 2018 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)  [Contact GitHub](#) [API](#) [Training](#) [Shop](#) [Blog](#) [About](#)

Figure 5.2. SPHERA on GitHub: first FOSS release (SPHERA v.8.0). Documentation.

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

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

AndreaAmicarelliRSE / SPHERA

Dismiss

Join GitHub today

GitHub is home to over 28 million developers working together to host and review code, manage projects, and build software together.

Sign up

SPHERA (RSE SpA): Smoothed Particle Hydrodynamics research software; mesh-less Computational Fluid Dynamics code.

174 commits1 branch1 release3 contributorsGPL-3.0

Branch: masterNew pull requestFind fileClone or download

AndreaAmicarelliRSE Set theme jekyll-theme-slateLatest commit fa6df0b on 13 Jun

bin	Minor modifications.	2 years ago
debug	SA-SPH. No-slip conditions are activated in 3D (Di Monaco et al., 201...	a year ago
debug_omp	SA-SPH. No-slip conditions are activated in 3D (Di Monaco et al., 201...	a year ago
doc	Minor modifications.	5 months ago
input	Minor modifications.	4 months ago
src	Minor modifications.	5 months ago
COPYING.txt	Minor modifications.	2 years ago
_config.yml	Set theme jekyll-theme-slate	2 months ago

Figure 5.4. SPHERA on GitHub: master branch or trunk (commit 13Jun2018).

AndreaAmicarelliRSE / SPHERA

Branch: master SPHERA / input /

Create new fileFind fileHistory

AndreaAmicarelliRSE Minor modifications. 1 Latest commit 3e5b2fa on 3 Apr

..		
Alpe_Gera	Minor modifications.	6 months ago
Alpe_Gera_dbf_Lanzada_substations	Minor modifications.	4 months ago
ICOLD_dam_breach_trunks	Minor modifications.	6 months ago
SPH_udb_Kim2015HYDROL	Minor modifications.	5 months ago
San_Fernando_Lower_van_Norman_...	Minor modifications. The following test cases are published at SPHERI...	2 years ago
dam_breach_ICOLD_DB-SPH_BD	Minor modifications.	6 months ago
dike_breach_2D_expSchHag12JHR_I...	Minor modifications. The following test cases are published at SPHERI...	2 years ago
edb_2D_FraCap02	Minor modifications.	6 months ago
edb_2D_FraCap02_Taipei	Minor modifications.	6 months ago
edb_2D_Spi05	Minor modifications.	6 months ago
edb_ICOLD	Minor modifications.	6 months ago
edb_KarlSand	Minor modifications.	6 months ago
edb_Pon10	Minor modifications.	6 months ago
rectangular_side_weir_Fr_0_491	Minor modifications.	6 months ago
release_v_8_0	Minor modifications.	2 years ago
sloshing_tank_TbyTn_1_07	Bug fixed. Restart. Arrays Z_fluid_max, q_max,	2 years ago
spherical_Couette_flows	Minor modifications. The following test cases are published at SPHERI...	2 years ago
tests	DB-SPH-NS-bl. Fast run in progress.	3 years ago
wave_motion_for_WaveSAX	Minor modifications.	6 months ago
SPHERA_main_input_file_template_...	Minor modifications.	4 months ago

Figure 5.5. SPHERA on GitHub: master branch. Input files (commit 13Jun2018).

AndreaAmicarelliRSE / SPHERA

Branch: master SPHERA / src /

Create new fileFind fileHistory

AndreaAmicarelliRSE Minor modifications. Latest commit 97320bc on 26 Feb

..		
BC	Minor modifications.	6 months ago
BE_mass	Minor modifications.	6 months ago
BE_momentum	Minor modifications.	6 months ago
Body_Transport	Minor modifications.	6 months ago
Constitutive_Equation	Minor modifications.	6 months ago
DB_SPH	Minor modifications.	6 months ago
Erosion_Criterion	Minor modifications.	6 months ago
Geometry	Minor modifications.	6 months ago
IC	Minor modifications.	5 months ago
Interface_dispersion	Minor modifications.	6 months ago
Main_algorithm	Minor modifications.	6 months ago
Modules	Minor modifications.	6 months ago
Neighbouring_Search	Minor modifications.	5 months ago
Post_processing	Minor modifications.	5 months ago
Pre_processing	Minor modifications.	5 months ago
SA_SPH	Minor modifications. Post_processing. Fix on the formula for vulnerab...	5 months ago
Strings	Minor modifications.	6 months ago
Time_Integration	Minor modifications.	6 months ago
Makefile	Minor modifications.	6 months ago
program_unit_section_labels.f90	Minor modifications.	6 months ago

Figure 5.6. SPHERA on GitHub: master branch. Source code (commit 13Jun2018).

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).

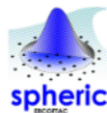


Figure 5.7. SPHERA indexed at SPHERIC web site (August 2018, extract from the list of the 13 SPH codes indexed at <http://spheric-sph.org/sph-projects-and-codes>).

6. THE SCHEME FOR TRANSPORT OF SOLID BODIES AND THE SEMI-ANALYTIC APPROACH AS A BOUNDARY TREATMENT SCHEME FOR FIXED BOUNDARIES

After 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. 6.7) and solid-solid (Sec.6.8) interactions.

The following sub-sections provide details on: improving 3D rotations (Sec.6.4); the sliding friction force (Sec.6.5); the body-boundary normal reaction force under sliding (at null normal velocity, Sec.6.6); the normal restitution coefficient (Sec.6.9).

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 ($\langle \cdot \rangle_I$) of a generic function (f) is defined as:

$$\langle f \rangle_{I, \underline{x}_0} \equiv \int_{V_h} f W dx^3 \quad (6.1)$$

where W is the kernel function ([10]), \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:

$$\left\langle \frac{\partial f}{\partial x_i} \right\rangle_{I, \underline{x}_0} = \int_{A_h} f W n_i dx^2 - \int_{V_h} f \frac{\partial W}{\partial x_i} dx^3 \quad (6.2)$$

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, [83]; Macia et al., 2012, [121]; Mayrhofer et al., 2013, [131]; Ferrand et al., 2013, [57]; Amicarelli et al., 2013, [6]).

Far from boundaries, the SPH particle approximation of (6.2) reads:

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

where a summation on particle volumes (ω) replaces the volume integral. The subscripts “ o ” 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, [203]; Amicarelli et al., 2015, [7]; Di Monaco et al., 2011, [46]); sloshing tanks (Souto-Iglesias et al., 2006, [188]; Delorme et al., 2009, [42]; Amicarelli et al., 2013, [6]); wave motion (e.g., Patel et al., 2009, [156]; Antuono et al., 2011, [14]; Liu et al., 2013, [118]; Omidvar et al., 2012a, [149]); hydraulic turbines (Marongiu et al., 2010, [125]); fast landslides (e.g., Kumar et al., 2013, [102]); erosion and bed-load transport (Manenti et al., 2012, [122]); liquid jets (e.g., Koukouviniis et al., 2013, [100]); pollutant dispersion; astrophysics (e.g., Price & Monaghan, 2007, [164]); magneto-fluid dynamics (e.g., Price, 2012, [162]); multi-phase and multi-fluid flows (e.g., Kajtar & Monaghan, 2012, [95]).

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

This section relies on [46] and [7], whose reading is 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, [210]). Its basic features are deeply described in Di Monaco et al. (2011, [46]) and here briefly reported.

Consider Euler's momentum and continuity equations, in the following forms:

$$\begin{aligned} \frac{du_i}{dt} &= -\frac{1}{\rho} \frac{\partial p}{\partial x_i} - \delta_{i3}g = -\frac{\partial \left(\frac{p}{\rho} \right)}{\partial x_i} - \frac{p}{\rho^2} \frac{\partial \rho}{\partial x_i} - \delta_{i3}g, \quad i=1,2,3 \\ \frac{d\rho}{dt} &= -\rho \nabla \cdot \underline{u} \end{aligned} \quad (6.4)$$

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"; [210]):

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

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 (" ϕ "). 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 [46]) hypothesizes the following linearization and assumptions to compute f in V_h :

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

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

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

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):

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

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:

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

where 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):

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

where \underline{a}_s represents a new acceleration term due to the fluid-body interactions, ν_M is the artificial viscosity (Monaghan, 2005, [138]), 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:

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

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 [7], whose reading 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:

$$\begin{aligned}
 \frac{d\underline{u}_{CM}}{dt} &= \frac{\underline{F}_{TOT}}{m_B} \\
 \frac{d\underline{x}_{CM}}{dt} &= \underline{u}_{CM} \\
 \underline{M}_{TOT} &= \underline{I}_C \frac{d\underline{\chi}_B}{dt} + \frac{d\underline{I}_C}{dt} \underline{\chi}_B = \underline{I}_C \frac{d\underline{\chi}_B}{dt} + \underline{\chi}_B \times \left(\underline{I}_C \underline{\chi}_B \right) \Rightarrow \frac{d\underline{\chi}_B}{dt} = \underline{I}_C^{-1} \left[\underline{M}_{TOT} - \underline{\chi}_B \times \left(\underline{I}_C \underline{\chi}_B \right) \right] \\
 \frac{d\underline{\alpha}}{dt} &= \underline{\chi}_B
 \end{aligned} \tag{6.12}$$

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 ($\underline{\chi}_B$ 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 axes and the global reference system). \underline{M}_{TOT} represents the associated torque acting on the body and \underline{I}_C the matrix of the moments of inertia of the computational body (Einstein’s notation works for the subscript “ $_i$ ”):

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

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:

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

\underline{G} represents the gravity force, whereas \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). In case of inertial and quasi-inertial fluid flows, we do not need to refer to neither turbulence scheme nor tangential stresses (simplifying hypothesis). Nonetheless, shear forces can be represented under more generic conditions (Sec.6.5).

Gravity is always active. The expressions “gravity deactivated” (input file template) only refers to the activation of drag and reaction forces, which temporarily balances gravity components, during

an impingement or in case of sliding. The approximations refer to drag and reaction forces, not to gravity.

The fluid-solid interaction is expressed by the following pressure force:

$$\underline{P}_F = \sum_s p_s A_s \underline{n}_s \quad (6.15)$$

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.6.7.

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:

$$\underline{M}_{TOT} = \sum_s \underline{r}_s \times \underline{F}_s \quad (6.16)$$

Time integration of (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 vector sum of the velocity of the corresponding body barycentre and the relative velocity:

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

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

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

$$\underline{R}_B = \underline{R}_x \underline{R}_y \underline{R}_z, \quad d\alpha_B = \underline{\chi}_B dt$$

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

6.4. Improving 3D rotations

The equations:

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

$$(6.19)$$

$$\underline{R}_B = \underline{R}_x \underline{R}_y \underline{R}_z$$

are not exact and represent a limit when working with the a rotation matrix based on Euler angles: the first equation of (6.19) is exact only for 1D rotations, the second one is never exact (non-commutative property of the rotation matrices based on Euler angles).

The following formula represents a generalized Euler-Newton equation for 3D rotations and provides exact results in 3D:

$$\frac{d\underline{\underline{R}}}{dt} = \underline{\underline{skew}}(\underline{\underline{\chi}}_B)\underline{\underline{R}} \quad (6.20)$$

Consider rotation matrices based on Euler's theorem ("each 3D rotation can be represented as a 1D rotation about a generic axis" of a unique rotation angle θ_R). This approach is also called "matrix to axis-angle" and is used by Rodrigues formula to define the rotation matrix:

$$\underline{\underline{R}} = \underline{\underline{I}} + (\sin \theta_R) \underline{\underline{skew}}(\hat{n}_R) + (1 - \cos \theta_R) \underline{\underline{skew}}^2(\hat{n}_R) \quad (6.21)$$

where the rotation axis is represented by the unit vector \underline{n}_R .

The cross-product matrix is a skew symmetric matrix:

$$\underline{\underline{skew}}(\hat{n}_R) \equiv \begin{bmatrix} 0 & -n_{R,3} & n_{R,2} \\ n_{R,3} & 0 & -n_{R,1} \\ -n_{R,2} & n_{R,1} & 0 \end{bmatrix} \quad (6.22)$$

The expansion of R_{ij} reads:

$$\underline{\underline{R}} = \begin{bmatrix} n_{R,1}^2 + (1 - n_{R,1}^2) \cos \theta_R & n_{R,1} n_{R,2} (1 - \cos \theta_R) - n_{R,3} \sin \theta_R & n_{R,1} n_{R,3} (1 - \cos \theta_R) + n_{R,2} \sin \theta_R \\ n_{R,1} n_{R,2} (1 - \cos \theta_R) + n_{R,3} \sin \theta_R & n_{R,2}^2 + (1 - n_{R,2}^2) \cos \theta_R & n_{R,2} n_{R,3} (1 - \cos \theta_R) - n_{R,1} \sin \theta_R \\ n_{R,1} n_{R,3} (1 - \cos \theta_R) - n_{R,2} \sin \theta_R & n_{R,2} n_{R,3} (1 - \cos \theta_R) + n_{R,1} \sin \theta_R & n_{R,3}^2 + (1 - n_{R,3}^2) \cos \theta_R \end{bmatrix} \quad (6.23)$$

The rotation angle depends on the angular velocity and the time step duration:

$$\hat{n}_R \theta_R = \underline{\underline{\chi}}_B dt$$

$$\hat{n}_R = \frac{\underline{\underline{\chi}}_B}{|\underline{\underline{\chi}}_B|} \quad (6.24)$$

$$\theta_R = |\underline{\underline{\chi}}_B| dt$$

When working with rotation matrices, Gimbal lock can be only prevented by Rodrigues formula.

So far, SPHERA only applies the approach described in this section to provide output data in order to initialize a following restarted simulation. In this frame, SPHERA determines the rotation matrix which provides a rotation from the initial time of the former simulation to a generic time for which output data are available for a following restarted simulation. This represents an approximated procedure which only guarantees a correct re-initialization of the relative position of the first body particle. As the body centre of mass is correctly imposed, the error in the body orientation (of a reinitialized simulation) is limited.

However, the approach of this section could be used in a more generic way to reproduce the exact body orientation in the body dynamics balance equations at every time step (Sec.6.3).

Hereafter is reported the procedure to determine the rotation matrix to move from a first vector (the relative position of the first body particle at the initial time) to a second vector (the relative position of the first body particle at a generic output time).

The rotation axis unit vector is estimated as follows:

$$\underline{\hat{n}}_R = \frac{\underline{v}_A \times \underline{v}_B}{|\underline{v}_A \times \underline{v}_B|} \quad (6.25)$$

The cross product is not sufficient to distinguish between θ_R and $(\pi - \theta_R)$:

$$\sin \theta_R = \left(\frac{|\underline{v}_A \times \underline{v}_B|}{|\underline{v}_A| |\underline{v}_B|} \right), \text{ but don't use } \theta_R = \arcsin \left(\frac{|\underline{v}_A \times \underline{v}_B|}{|\underline{v}_A| |\underline{v}_B|} \right) \quad (6.26)$$

However, consider that θ_R from the cross product automatically and correctly provides an angle between 0 and π ($\sin(\theta_R)$ is non-negative).

The sine and cosine functions of the rotation angle read:

$$\sin \theta_R = \frac{|\underline{v}_A \times \underline{v}_B|}{|\underline{v}_A| |\underline{v}_B|}, \quad \cos \theta_R = \frac{\underline{v}_A \cdot \underline{v}_B}{|\underline{v}_A| |\underline{v}_B|} \quad (6.27)$$

Provided its sine and cosine functions, the value of the rotation angle can be estimated by means of the “atan2” function:

$$\theta_R = \text{ATAN}_2(\sin \theta_R, \cos \theta_R) = \begin{cases} \text{ATAN} \left(\frac{\sin \theta_R}{\cos \theta_R} \right), & \cos \theta_R > 0 \\ \pi + \text{ATAN} \left(\frac{\sin \theta_R}{\cos \theta_R} \right), & \cos \theta_R < 0 \\ \frac{\pi}{2}, & \cos \theta_R = 0, \quad \sin \theta_R > 0 \\ -\frac{\pi}{2}, & \cos \theta_R = 0, \quad \sin \theta_R < 0 \end{cases} \quad (6.28)$$

Once obtained the rotation axis and angle, Rodrigues formula is used to build the rotation matrix and rotate the whole body by means of a single rotation, from the initial time to the time chosen.

6.5. Sliding friction force

6.5.1. Aerial stage (non-negative value for the input friction angle and body-frontier interactions)

The simulated and exact formulation (*) for the sliding friction force are equal:

$$\underline{T}_S^* = -\underline{G} \cdot \underline{n}_w \mu_{sf} \underline{s}_w = -\underline{G} \cdot \underline{n}_w \tan(\varphi_{dry}) \underline{s}_w \quad (6.29)$$

where φ is the sliding friction angle, the subscript “*dry*” refers to dry conditions, \underline{s}_w is the unit vector parallel to the local frontiers and with direction opposite to the velocity vector of the body barycentre (projected on the local DEM). The overall normal of the neighbouring frontiers is the unit vector aligned with the vector sum of the neighbouring normals.

The coefficient of sliding friction (μ_{sf}) reads:

$$\mu_{sf} = \tan(\varphi_{dry}) \quad (6.30)$$

The absolute value for the exact formulation reads:

$$|T_S^*| = mg \cos(\alpha_{DTM}) \tan(\varphi_{dry}) \quad (6.31)$$

where the cosine of the slope angle is obtained by means of a dot product:

$$\cos \alpha_{DTM} = |\underline{n}_w \cdot \underline{k}| \quad (6.32)$$

The direction of the sliding friction force is the opposite to the velocity direction of the centre of mass of the computational body.

The following limiter applies to the sliding friction force:

$$|T_S^*|_{\max} = \frac{|\underline{u}_{\tan}| m}{dt} \quad (6.33)$$

where \underline{u}_{\tan} is the maximum particle velocity (all over the computational body) tangential to the interacting local frontier elements.

The current code versions assumes the following approximations: a unique friction angle applies to all the body-frontier interactions; the vector sum of the normal reaction force under sliding and the sliding friction force provide no contribution to the body torque (nevertheless the limiter for the sliding friction force depends on the velocity of the solid particles interacting with the frontiers).

6.5.2. Aerial stage (negative value for the input friction angle or body-body interactions)

The sliding friction force (\underline{T}_S : drag force on a body under body-boundary interactions) is approximately represented by means of a tangential force, which balances the tangential component of gravity:

$$\underline{T}_S = -[\underline{G} - (\underline{G} \cdot \underline{n}_w) \underline{n}_w] \quad (6.34)$$

Its absolute value is:

$$|T_S| = mg \sin(\alpha_{DTM}) \quad (6.35)$$

where α_{DTM} is the slope angle and \underline{k} the unit vector aligned with the vertical axis:

$$\sin \alpha_{DTM} = |\underline{n}_w \times \underline{k}| \quad (6.36)$$

The removal of the gravity force component which is parallel to the bottom is equivalent to introducing an approximated sliding friction force, where the slope angle approximates the sliding friction angle.

$$\begin{aligned} \underline{T}_S &= \underline{T}_S^* \Rightarrow \\ &\Rightarrow mg \sin \alpha_{DTM} = mg \cos(\alpha_{DTM}) \tan(\varphi_{dry}) \Rightarrow \end{aligned} \quad (6.37)$$

$$\Rightarrow \varphi_{dry} \cong \alpha_{DTM}$$

This approximation might be acceptable, especially in case the sliding friction angle is unavailable.

6.5.3. Submerged stage

The sliding friction force is negligible (hypothesis on the inertial fluid flows, Amicarelli et al., 2015, CAF):

$$\underline{T}_S = \underline{0} \quad (6.38)$$

The exact formulation reads:

$$\underline{T}_S^* = -(\underline{G} + \underline{P}_F) \cdot \underline{n}_w \tan(\varphi_{wet}) \underline{s}_w \quad (6.39)$$

6.6. Body-boundary normal reaction force under sliding (at null normal velocity)

6.6.1. Aerial stage

The exact formulation is correctly represented:

$$\underline{P}_S = \underline{P}_S^* = -(\underline{G} \cdot \underline{n}_w) \underline{n}_w, \quad \underline{u} \cdot \underline{n}_w = 0 \quad (6.40)$$

This term is added to the body-boundary normal force in the main file of this documentation -Eq.(6.32)-. The overall normal of the neighbouring frontiers is the unit vector aligned with the vector sum of the neighbouring normals.

6.6.2. Submerged stage

The normal reaction is formally null.

$$\underline{P}_S = \underline{0}, \quad \underline{u} \cdot \underline{n}_w = 0 \quad (6.41)$$

According to Monaghan (2005), the normal forces of the boundary force particles (6.32) dynamically restore the normal reaction force (body-boundary interactions), despite some body spurious oscillations (normal to the frontier) in the interaction zone (noise amplitude comparable with the spatial resolution).

$$\underline{P}_S^* = -(\underline{G} + \underline{P}_F) \cdot \underline{n}_w, \quad \underline{u} \cdot \underline{n}_w = 0 \quad (6.42)$$

6.7. Fluid-body interaction terms

This section relies on [7], whose reading is suggested for further details.

The fluid-body interaction terms rely on the boundary technique introduced by Adami et al. (2012, [1]), implemented and adapted for free-slip conditions (Amicarelli et al., 2015, [8]). 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):

$$C_s = 2\rho_0 \sum_s [(\underline{u}_s - \underline{u}_0) \cdot \underline{n}_s] \underline{W}_s \omega_s \quad (6.43)$$

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

$$\underline{a}_s = \sum_s \left(\frac{p_s + p_0}{\rho_0^2} \right) W_s' m_s \quad (6.44)$$

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):

$$\left(\frac{d\underline{u}_f}{dt} \right) \cdot \underline{n}_w = \left(-\frac{1}{\rho_f} \nabla p_f + \underline{g} \right) \cdot \underline{n}_w = \underline{a}_w \cdot \underline{n}_w \quad (6.45)$$

The “wall” acceleration at the position of a generic body particle can then be derived by linearizing (6.45). This depends on the particular computational fluid particle “ o ” we are considering, so that we can refer to the interaction subscript “ s,o ”:

$$\begin{aligned} \int_0^s \left(-\frac{1}{\rho_f} \nabla p_f \right) \cdot \underline{n}_w d\underline{l}_n &= \int_0^s (\underline{a}_w - \underline{g}) \cdot \underline{n}_w d\underline{l}_n \Rightarrow \\ \Rightarrow p_{s,o} &\approx p_0 + \rho_0 [(\underline{g} - \underline{a}_s) \cdot \underline{n}_w] [(\underline{x}_s - \underline{x}_o) \cdot \underline{n}_w] \end{aligned} \quad (6.46)$$

where $d\underline{l}_n$ is a vectorial length element along the centreline of the two particles, projected along the wall element normal.

One may apply a SPH interpolation over all the pressure values estimated according to Adami et al. (2012, [1]) to derive a unique pressure value for a body particle:

$$p_s = \frac{\sum_0 p_{s,o} W_{os} \left(\frac{m_0}{\rho_0} \right)}{\sum_0 W_{os} \left(\frac{m_0}{\rho_0} \right)} = \frac{\sum_0 [p_0 + \rho_0 [(\underline{g} - \underline{a}_s) \cdot \underline{n}_w] [(\underline{x}_s - \underline{x}_o) \cdot \underline{n}_w]] W_{os} \left(\frac{m_0}{\rho_0} \right)}{\sum_0 W_{os} \left(\frac{m_0}{\rho_0} \right)} \quad (6.47)$$

In case of no-slip conditions, (6.47) assumes the following form:

$$p_s = \frac{\sum_0 p_{s,o} W_{s0} \left(\frac{m_0}{\rho_0} \right)}{\sum_0 W_{s0} \left(\frac{m_0}{\rho_0} \right)} = \frac{\sum_0 [p_0 + \rho_0 (\underline{g} - \underline{a}_s) \cdot \underline{r}_{s0}] W_{s0} \left(\frac{m_0}{\rho_0} \right)}{\sum_0 W_{s0} \left(\frac{m_0}{\rho_0} \right)} \quad (6.48)$$

The pressure value of (6.47) or (6.48) is finally used in (6.45).

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 (6.48) -“ s_o ” interaction-. If the on-going body particle “ s ” belongs to the body surface, then it is immediately considered as representative. Otherwise, the fluid particle “ o ” 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 “ θ ” and “ s ”, is then selected. This particle provides the normal “ \underline{n}_s ” for the fluid-solid particle interaction “ $s\theta$ ” in (6.48). The assumption (6.45) 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 (6.48). Nevertheless, we prefer defining a maximum threshold for $|\underline{a}_s|$, here equal to $10g$.

Two pressure limiters can be activated to treat the fluid-solid interface.

The “negative value pressure limiter” provides a minimum threshold:

$$p_{\min} = 0 \quad (6.49)$$

The “positive value pressure limiter” acts as a LPRS and provides a maximum pressure threshold:

$$p_{\max,s,body} = \gamma(z_{FS,\max} - z_{s,\min,body} + 2h) + (1.05\rho_{ref})c_{ref}|\underline{u}_{s,\max,body} - \underline{u}_{\max}| \quad (6.50)$$

The maximum and minimum operators in the Right Hand Side of (6.50) apply to the whole domain.

6.8. Solid-solid interaction terms

This section relies on [7], whose reading is suggested for further details.

The solid-solid interaction term in (6.14) \underline{P}_s represents body-body and body-boundary 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, [138]) 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 :

$$\underline{a}_{bfp,jk} = \frac{f_{bfp}}{r_{jk}} \frac{m_k}{m_j + m_k} \underline{r}_{jk} \quad (6.51)$$

The analytic function f_{bfp} is symmetric with respect to the impact point. The dependence of (6.51) on the particle masses allows conserving both global momentum ($m_j \underline{a}_{bfp,jk} = -m_k \underline{a}_{bfp,kj}$) and kinetic energy (one may notice that $\underline{r}_{jk} = -\underline{r}_{kj}$ 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:

$$\underline{P}_s = \sum_K \underline{P}_{BK} + \sum_{K^*} \underline{P}_{BK^*} \quad (6.52)$$

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 ”:

$$\underline{P}_{BK} = -\alpha_I \sum_j \sum_k \frac{2u_{\perp,jk}^2}{r_{per,jk}} \frac{m_j m_k}{m_j + m_k} \Gamma_{jk} \left(1 - \frac{r_{par,jk}}{dx_s} \right) \underline{n}_k \quad (6.53)$$

The components of the inter-particle relative distance, r_{par} and r_{per} , are parallel and perpendicular to the neighbour normal, respectively. The term within brackets in (6.53) 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, [138]) and depends on $q = r_{jk}/h$:

$$\Gamma_{jk} = \begin{cases} \frac{2}{3}, & 0 \leq q < \frac{2}{3} \\ \left(2q - \frac{3}{2}q^2 \right), & \frac{2}{3} \leq q < 1 \\ \frac{1}{2}(2-q)^2, & 1 \leq q < 2 \\ 0, & 2 \leq q \end{cases} \quad (6.54)$$

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, [138]) 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:

$$u_{\perp,jk}(t) = \max_{j,k,t^*} \left\{ \left| (\underline{u}_j - \underline{u}_k) \cdot \underline{n}_k \right| \right\}, \quad t_0 \leq t^* \leq t \quad (6.55)$$

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, (6.55) 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, (6.53) 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 , (6.53) 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:

$$P_{BK}' = \frac{2u_{\perp,BK}^2}{r_{per,BK}} \frac{m_B m_K}{m_B + m_K} \Gamma_{BK}, \quad r_{per,BK} = \min_{B,K} \{r_{per,jk}\}, \quad u_{\perp,BK}^2 = \max_{B,K} \{u_{\perp,jk}^2\} \quad (6.56)$$

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}' :

$$\alpha_I = \frac{\frac{u_{\perp,BK}^2}{r_{per,BK}} \frac{m_B m_K}{m_B + m_K} \Gamma_{BK}}{\sum_j \sum_k \left[\frac{u_{\perp,jk}^2}{r_{per,jk}} \frac{m_j m_k}{m_j + m_k} \Gamma_{jk} \left(1 - \frac{r_{par,jk}}{dx_s} \right) \right]} \quad (6.57)$$

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

$$\alpha_I = \frac{\frac{1}{r_{per,BK}} \frac{m_B m_K}{m_B + m_K} \Gamma_{BK}}{\sum_j \sum_k \left[\frac{1}{r_{per,jk}} \frac{m_j m_k}{m_j + m_k} \Gamma_{jk} \left(1 - \frac{r_{par,jk}}{dx_s} \right) \right]} \quad (6.58)$$

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 (6.57) 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):

$$\underline{P}_{BK^*} = -\alpha_I \sum_j \frac{2u_{\perp,jK^*}^2}{r_{per,jK^*}} m_j \Gamma_{jK^*} \underline{n}_{K^*}, \quad \alpha_I = \frac{\frac{m_B}{r_{per,BK^*}} \Gamma_{BK^*}}{\sum_j \left(\frac{m_j}{r_{per,jK^*}} \Gamma_{jK^*} \right)} \quad (6.59)$$

6.9. Normal restitution coefficient

The simulated normal restitution coefficient (R_n) is equal to unity if all the following conditions are satisfied: homogeneous velocity of the solid particles belonging to the impinging body, impingement with a single frontier element, isolated impingement, body axes aligned with the frontier axes. Otherwise (real cases), the scheme for body-boundary force is dissipative and represents $R_n < 1$. The equivalent value of the restitution coefficient might be estimated a posteriori.

7. THE SCHEME FOR DENSE GRANULAR FLOWS

This section describes the mathematical and numerical models for dense granular flows (Amicarelli et al., 2017, [9], Sec.7.1) and its possible speed-up by means of a 2-interface 3D erosion scheme (Amicarelli & Agate, 2014, [5], Sec.7.2), which extends the (1-interface) 2D erosion scheme of Manenti et al. (2012, [122], Sec.7.2).

This mixture model for dense granular flows (e.g., bed-load transport, fast landslides) is consistent with the “packing limit” of the Kinetic Theory of Granular Flow (KTGF) and no tuning parameter is used to represent the mixture viscosity.

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. Mixture model for dense granular flows

This whole section constantly refers to Amicarelli et al. (2017, [9]), but some further details and more recent code updates.

This SPH model represents the mixture of pure fluid and non-cohesive solid granular material, under the “packing limit” of the Kinetic Theory of Granular Flow (KTGF, [16]) for dense granular flows. This limit refers to the maximum values of the solid phase volume fraction and is peculiar of bed-load transport (e.g., erosional dam breaks) and fast landslides.

The continuity equation for the fluid phase (“f”) can be expressed as follows ([16]):

$$\frac{d(\rho_f \varepsilon_f)}{dt} = - \frac{\partial(\rho_f \varepsilon_f u_{f,j})}{\partial x_j} \quad (7.1)$$

where ρ ($\text{kg} \times \text{m}^{-3}$) is density, ε the phase volume fraction, \underline{u} ($\text{m} \times \text{s}^{-1}$) the velocity vector, t represents time and \underline{x} (m) the position vector. Einstein’s notation applies to the subscript “j”, hereafter.

The continuity equation for the incompressible solid phase (“s”) can be expressed as follows ([16]):

$$\frac{d(\rho_s \varepsilon_s)}{dt} = - \frac{\partial(\rho_s \varepsilon_s u_{s,j})}{\partial x_j} \quad (7.2)$$

The volume balance equation assumes the following form:

$$\varepsilon_s + \varepsilon_f = 1 \quad (7.3)$$

After defining the mixture density and velocity (the subscript “m” is always omitted):

$$\rho = \rho_f \varepsilon_f + \rho_s \varepsilon_s, \quad u_i \equiv \frac{\rho_f \varepsilon_f u_{f,i} + \rho_s \varepsilon_s u_{s,i}}{\rho} \quad (7.4)$$

the summation of (7.1) and (7.2) provides:

$$\frac{d\rho}{dt} = - \frac{\partial(\rho u_j)}{\partial x_j} \quad (7.5)$$

The model assumes that SPH particles are conservative (i.e. mixture particles do not exchange net mass fluxes with the surrounding environment), which is a reasonable hypothesis for high solid volume fractions in saturated soils, according to the “packing limit” of the Kinetic Theory of Granular Flow (KTGF, [16]):

$$\frac{d\rho}{dt} = 0 \Rightarrow \frac{\partial u_j}{\partial x_j} = 0 \quad (7.6)$$

Starting from (7.6), the model adopts a Weakly Compressible approach to obtain:

$$\frac{d\rho}{dt} = -\rho \frac{\partial u_j}{\partial x_j} \quad (7.7)$$

Following the multi-phase approach of [127], the SPH approximation of (7.7) can be expressed as follows:

$$\frac{d\rho_0}{dt} = \rho_0 \sum_b (u_{b,j} - u_{0,j}) \frac{\partial W}{\partial x_j} \Big|_b \omega_b + 2\rho_0 \int_{V_h} [(\underline{u}_w - \underline{u}_0) \cdot \underline{n}] n_j \frac{\partial W}{\partial x_j} dx^3 \quad (7.8)$$

where \underline{n} is the unit vector normal to the frontier. The subscripts “ 0 ”, “ b ” and “ w ” refer to the computational particle, a neighbouring particle and a wall frontier, respectively. The integral boundary term is computed according to [46] and represents the effects of wall frontiers.

Considering the KTGF, the momentum equation for the fluid phase can be expressed as follows ([16]):

$$\frac{d(\rho_f \varepsilon_f u_{f,i})}{dt} = -\delta_{i3} g \rho_f \varepsilon_f - \varepsilon_f \frac{\partial p_f}{\partial x_i} + \frac{\partial \tau_{f,ij}}{\partial x_j} - K_{gs} (u_{f,i} - u_{s,i}) \quad (7.9)$$

where τ_{ij} (Pa) is the deviatoric (or shear) stress tensor, g ($\text{m} \times \text{s}^{-2}$) gravity acceleration and p (Pa) pressure. The last term depends on the relative velocity between the phases (filtration process) through the drag coefficient K_{gs} ($\text{kg} \times \text{m}^{-3} \times \text{s}^{-1}$).

The momentum equation for the solid phase can be expressed as follows ([16]):

$$\frac{d(\rho_s \varepsilon_s u_{s,i})}{dt} = -\delta_{i3} g \rho_s \varepsilon_s - \frac{\partial p_s}{\partial x_i} + \frac{\partial \tau_{s,ij}}{\partial x_j} - \varepsilon_s \frac{\partial p_f}{\partial x_j} + K_{gs} (u_{f,i} - u_{s,i}) \quad (7.10)$$

Provided the volume equation and the definitions of the mixture velocity and density, the sum of (7.9) and (7.10) provides:

$$\frac{d(\rho u_i)}{dt} = -\delta_{i3} g \rho - \frac{\partial p_f}{\partial x_i} - \frac{\partial p_s}{\partial x_i} + \frac{\partial \tau_{f,ij}}{\partial x_j} + \frac{\partial \tau_{s,ij}}{\partial x_j} \quad (7.11)$$

Considering the assumption on conservative SPH particles, the shear stress gradient term of the fluid phase can be expressed as follows ([16]):

$$\frac{\partial \tau_{f,ij}}{\partial x_j} = \varepsilon_f \left[\frac{\mu_f}{3} \frac{\partial}{\partial x_i} \left(\frac{\partial u_{f,j}}{\partial x_j} \right) + \mu_f \frac{\partial^2 u_{f,i}}{\partial x_j^2} \right] \quad (7.12)$$

where μ ($\text{Pa} \times \text{s}$) represents viscosity. Under the hypothesis of plain strain, the shear stress gradient term is represented by [174] (visco-plastic model for dry granular material based on internal friction), by means of a parameter, which KTGF names frictional viscosity, as described in the following.

The pressure of the solid phase is treated as follows:

$$-\frac{\partial p_{s,m}}{\partial x_i} = -\frac{\partial \sigma'_m}{\partial x_i}, \quad \sigma'_m \equiv \sum_{i=1}^3 \frac{\sigma'_i}{3} \quad (7.13)$$

where σ'_i (Pa) are the effective stresses along the principal directions and σ'_m (Pa) is the mean effective stress ([174] refers to a smoothed approximation of the 3D Mohr-Coulomb criterion; the extension to Mohr-Coulomb-Terzaghi criterion for saturated soils is straightforward). The shear stress gradient term in the momentum equation can be expressed as follows ([174]):

$$\frac{\partial \tau_{s,ij}}{\partial x_j} = -k \frac{\partial}{\partial x_j} \left(\sigma_m' |e_{ij}|^{-1} (-e_{ij}) \right), \quad e_{ij} \equiv \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (7.14)$$

$$|e_{ij}| \equiv \left(\sum_{i,j} e_{ij}^2 \right)^{\frac{1}{2}} = \sqrt{2I_2(e_{ij})}, \quad k \equiv \sqrt{2}(\sin \varphi)$$

where φ (rad) is the internal friction angle, e_{ij} (s^{-1}) is the strain-rate tensor and $I_2(e_{ij})$ (s^{-2}) represents its second invariant (formulation for incompressible fluids). One may notice that the term (7.14) is potentially unstable at high internal friction angles and that, in the “packing limit” of the KTGF, the shear stress terms of the collisional-kinetic regime are zeroed. Eq.8.14 can be rearranged as follows:

$$\frac{\partial \tau_{s,ij}}{\partial x_j} = \sqrt{2}(\sin \varphi) \frac{\partial}{\partial x_j} \left(\frac{\sigma_m'}{\sqrt{2I_2(e_{ij})}} e_{ij} \right) \quad (7.15)$$

The strain-rate tensor reads:

$$e_{ij} \equiv \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (7.16)$$

whereas its second invariant (in case of free divergence flows) assumes the following expression:

$$I_2(e_{ij}) = \frac{1}{2} \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 + \left(\frac{\partial w}{\partial z} \right)^2 \right] + \frac{1}{4} \left[\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right)^2 \right] \quad (7.17)$$

Renormalization (Randles & Libertsy, 1996, [167]) applies to the velocity derivatives in (7.17), only for 2D simulations:

$$\left\langle \frac{\partial f}{\partial x_i} \right\rangle \equiv \sum_b (f_b - f_0) \left(\underline{B}_0 \cdot \underline{\nabla} W_b \right)_i \omega_b, \quad B_{0,i}^{-1} \equiv \sum_b (x_b - x_0)_j \frac{\partial W}{\partial x_i} \omega_b \quad (7.18)$$

The KTGF introduces the following definition for the frictional viscosity μ_{fr} (Pa·s):

$$\mu_{fr} \equiv \left(\frac{\sigma_m'(\sin \varphi)}{2\sqrt{I_2(e_{ij})}} \right) \quad (7.19)$$

to obtain:

$$\frac{\partial \tau_{s,ij}}{\partial x_j} = \frac{\partial}{\partial x_j} (2\mu_{fr} e_{ij}) \quad (7.20)$$

Eq.(7.20) is consistent with internal friction, as it does not depend on the magnitude of the strain-rate tensor.

In analogy to the Navier-Stokes equation (for incompressible flows), under the strong but accepted hypothesis of smooth spatial variations of μ_{fr} , (7.20) provides:

$$\frac{\partial \tau_{s,ij}}{\partial x_j} = \mu_{fr} \frac{\partial^2 u_i}{\partial x_j^2} \quad (7.21)$$

which is valid in the “packing limit” of the KTGF (i.e. for ε_s close enough to the value of 0.59, which is the maximum attainable volume fraction for a sheared inelastic hard sphere fluid, [103]). The model then defines the mixture total pressure p (Pa) as:

$$p = p_f + \sigma'_m \quad (7.22)$$

One may consider that the mean effective stress can only be formulated under simplifying assumptions (e.g., x , y and z need to be the principal axes). Thus, σ'_m is computed as the difference between the total pressure and the fluid pressure:

$$\sigma'_m = \max(p - \max(p_f, 0), 0) \quad (7.23)$$

Both fluid and solid pressures are limited to positive values as soils, which are either fully saturated or dry, do not bear tension. Considering the continuity equation, the momentum equation for the mixture can be rearranged as:

$$\frac{d(\rho u_i)}{dt} = -\delta_{i3} g \rho - \frac{\partial p_f}{\partial x_i} - \frac{\partial \sigma'_m}{\partial x_i} + \varepsilon_f \mu_f \frac{\partial^2 u_{f,i}}{\partial x_j^2} + H(\varepsilon_s - \varepsilon_{s,p}) \mu_{fr} \frac{\partial^2 u_{s,i}}{\partial x_j^2} \quad (7.24)$$

where $\varepsilon_{s,p} = \text{ca.} 0.59$ and $H(x)$ is the Heaviside step function:

$$H(x) = \frac{d}{dx} \max\{x, 0\} \quad (7.25)$$

Assuming SPH conservative particles implies that the velocity of each phase is basically equal to the one of the mixture:

$$u_{f,i} \cong u_i, \quad u_{s,i} \cong u_i \quad (7.26)$$

Considering (7.22) and the assumption of SPH conservative particles, (7.24) reduces to:

$$\frac{du_i}{dt} = -\delta_{i3} g - \frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j^2} \quad (7.27)$$

where the mixture viscosity μ is finally defined as:

$$\mu \equiv \varepsilon_f \mu_f + H(\varepsilon_s - \varepsilon_{s,p}) \mu_{fr} \quad (7.28)$$

and $\nu \equiv \frac{\mu}{\rho}$ ($\text{m}^2 \times \text{s}^{-1}$) is the mixture kinematic viscosity.

Following the multi-phase approach of [127], with the boundary treatment method proposed by [46], the SPH approximation of (7.27) becomes:

$$\begin{aligned} \left\langle \frac{du_i}{dt} \right\rangle_0 = & -\delta_{i3} g + \frac{1}{\rho_0} \sum_b (p_b + p_0) \frac{\partial W}{\partial x_i} \Big|_b \omega_b + 2 \frac{p_0}{\rho_0} \int_{V_h} \frac{\partial W}{\partial x_i} \Big|_b dx^3 + 2\nu \sum_b \frac{m_b}{\rho_0 r_{0b}} (\underline{u}_b - \underline{u}_0) \frac{\partial W}{\partial r} \Big|_b + \\ & -\nu_M \sum_b \frac{m_b}{\rho_0 r_{0b}^2} (\underline{u}_b - \underline{u}_0) \cdot (\underline{x}_b - \underline{x}_0) \frac{\partial W}{\partial x_i} \Big|_b - \nu_M (\underline{u}_{SA} - \underline{u}_0) \cdot \left(\int_{V_h} \frac{1}{r_{0w}^2} (\underline{x} - \underline{x}_0) \frac{\partial W}{\partial x_i} dx^3 \right) + 2\nu (\underline{u}_{w,i} - \underline{u}_{0,i}) \cdot \left(\int_{V_h} \frac{1}{r_{0w}} \frac{\partial W}{\partial r} dx^3 \right) \end{aligned} \quad (7.29)$$

where r (m) is the distance between two interacting particles, whereas ν_M ($\text{m}^2 \times \text{s}^{-1}$) stands for the artificial viscosity ([138]). The boundary value \underline{u}_{SA} ($\text{m} \times \text{s}^{-1}$) of the velocity in the external portion V_h' (m^3) of the kernel support is assigned according to [46]. So far, the last term, representing the bottom drag, has been validated in 2D.

The artificial viscosity is always activated, both for approaching and separating particles (the latter configuration was not considered in Di Monaco et al., 2011, [46]):

$$v_M = \frac{\alpha hc}{\rho} \quad (7.30)$$

Despite its formulation as a mono-phase mixture, the model needs to adopt a simplified approach to represent fluid pressure in the granular material. This parameter can be related to two different soil conditions: uniform fully saturated soil and uniform dry soil (the first condition being applied to all the test cases in this study):

$$p_f = \begin{cases} p_{f,blt-top} + \rho_f g(z_{blt-top}|_{x_0,y_0} - z_0) \cos^2(\alpha_{TBT}), & \text{fully saturated soil} \\ 0, & \text{dry soil} \end{cases} \quad (7.31)$$

where the subscript “ $_{blt-top}$ ” refers to the top of the bed-load transport layer (or the layer of saturated material). Eq.(7.31) assumes a 1D filtration flow parallel to the slope of the granular material. This simplifying hypothesis is still consistent with SPH conservative particles; α_{TBT} (rad) is the topographic angle at the top of the bed-load transport layer and lies between the local interface normal $\underline{n}_{blt-top}$ and the vertical:

$$\alpha_{TBT} = \max\left(\arccos(n_{blt-top,3}), \frac{\pi}{2}\right) \quad (7.32)$$

The angle limiter in (7.32) allows one to assign null p_f values in case of slope anomalies (very rare and unstable).

The mixture pressure is computed by means of a barotropic equation of state (linearized around a reference state indicated by subscript “ $_{ref}$ ”):

$$p \cong c_{ref}^2 (\rho - \rho_{ref}) \quad (7.33)$$

It has been shown that the artificial speed of sound c (m/s) in the Weakly-Compressible approach should be at least 10 times greater than the maximum velocity to reduce the pressure error associated to artificial compressibility effects below 1% ([127]). A unique speed of sound can be chosen (i.e. the highest among the SPH particle values, no matter about their phase volume fractions).

In order to reduce the computational time and avoid the unbounded growth of (7.19), a threshold for the mixture viscosity can be defined (v_{max}). Mixture particles with a higher viscosity are considered in the elasto-plastic regime of soil deformation and are kept fixed, whereas their pressure is derived from the mixture particles flowing above them. The threshold value is assumed to be high enough not to influence the simulation. At a fixed time step, v_{max} does not influence even the computational time (the Courant-Friedrichs-Lewy CFL number criterion dominates over the viscous term criterion), if the following condition is satisfied:

$$v_{max} \leq \frac{C_v hc}{CFL} \quad (7.34)$$

It follows that the time step duration is more probably ruled by either the CFL stability criterion, if the spatial resolution is coarse enough, or the viscous term stability criterion, if the spatial resolution is fine enough. In this case, it might be convenient to adopt a multi-step approach, where the time integration of the equations of motion for the fluid particles would be obtained with a longer time step than the one needed for the mixture particles.

No-slip conditions are suggested to be imposed on solid walls for 2D simulations at very fine spatial resolutions. The associated solid boundaries are in general in contact with the bottom of the fixed bed, so the choice of a no-slip rather than a slip condition did not play any role. On the other hand, for 3D simulations imposing free-slip conditions is suggested. In fact, the depth of the granular material layer is generally high enough and the interactions with solid walls quite exclusively concern either fluid particles at high Reynolds number or mixture particles with null

velocities, so that the choice of a slip condition everywhere appeared to be an appropriate compromise. Nevertheless, no-slip conditions should be in general applied to those mobile mixture particles which are interested by a locally laminar regime. This issue, which plays a minor role in the test cases of Amicarelli et al. (2017, [9]), represents a matter of on-going developments.

The present model does not need any tuning for the mixture viscosity. The only case-dependent numerical parameters refer to the spatial resolution (dx, h) and possibly to CFL number.

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:

$$K = \rho \frac{\partial p}{\partial \rho} = \rho c_{ref}^2 = \rho (A_{WC} U_{scale})^2 \quad (7.35)$$

The sound speed (c_{ref}) should be at least 10 times higher than the maximum velocity in the fluid (WC approach). This position (constant A_{WC} equal to 10) provides a maximum relative error on density of 1%, whereas the assumption $A_{WC}=4.5$ increases the density relative error to 5% (Monaghan, 2005).

The velocity scale U_{scale} reads:

$$U_{scale} = \max(|u|_{max}, \sqrt{2gY_{max}}) \quad (7.36)$$

where Y_{max} is the maximum water depth and $|u|_{max}$ is the maximum absolute value of velocity (the maxima operate both over the whole simulated time and the whole 3D domain space).

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, [122]), 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, [208]) reads:

$$\vartheta_c = \begin{cases} 0.1, & Re_* < 1 \\ 0.010595 * \ln(Re_*) + \frac{0.110476}{Re_*} + 0.0027197, & 1 \leq Re_* \leq 500 \\ 0.068, & Re_* > 500 \end{cases} \quad (7.37)$$

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

$$Re_* \stackrel{\text{def}}{=} \frac{ku_*}{\nu} \quad (7.38)$$

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, [122]) and those associated to the similarity theory or the SNBL:

$$z_0 = 0.11 \frac{v}{u_*} + \frac{k}{30}, \quad u_* = \frac{k_v U}{\ln\left(\frac{z}{z_0}\right)} \quad (7.39)$$

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:

$$U = \frac{zu_*^2}{v} \Rightarrow u_* = \sqrt{\frac{Uv}{z}} \quad (7.40)$$

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:

$$\vartheta \stackrel{\text{def}}{=} \frac{\tau}{(\gamma_s - \gamma)d}, \quad \tau = \rho u_*^2 \quad (7.41)$$

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:

$$\vartheta_{c,\beta\gamma} = k_{\beta\gamma} \vartheta_{c,00}, \quad 0 \leq k_{\beta\gamma} \leq k_{\beta 0} \quad (7.42)$$

$k_{\beta\gamma}$ is always non-negative and smaller than (or equal to) its 2D value $k_{\beta 0}$ ($\gamma=0$). 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 ($\beta < 0$), $k_{\beta\gamma}$ can be higher than the unity. In this case, (7.42) 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:

$$\underline{n}_{int,0} = \frac{\sum_{bf} (\underline{x}_{bf} - \underline{x}_0) W_{bf} \omega_{bf}}{|\sum_{bf} (\underline{x}_{bf} - \underline{x}_0) W_{bf} \omega_{bf}|} \quad (7.43)$$

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):

$$\beta = \arcsin(-u_{b,3}) \quad (7.44)$$

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:

$$\beta = -\left[\frac{\pi}{2} - \arcsin(n_{int,0,z})\right] \text{sign}(u_{f,s,3}), \quad \underline{u}_{f,s} = \underline{u}_f - \underline{u}_f \cdot \underline{n}_{int} \quad (7.45)$$

The transversal slope angle γ is defined as:

$$\gamma = \arcsin|n_{2,z}|, \quad \underline{n}_2 = \underline{n}_{int} \wedge \underline{u}_f \quad (7.46)$$

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, [177]):

$$\begin{aligned} ak_{\beta\gamma}^2 + bk_{\beta\gamma} + c &= 0 \Rightarrow k_{\beta\gamma} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \\ a &\stackrel{\text{def}}{=} (1 - \Delta), b \stackrel{\text{def}}{=} 2 \left\{ \frac{\Delta}{\sqrt{1 + \tan^2 \beta + \tan^2 \gamma}} + \frac{\sin \beta}{\tan \varphi} \right\}, \\ c &\stackrel{\text{def}}{=} \frac{1 + \Delta}{1 + \tan^2 \beta + \tan^2 \gamma} \left(-1 + \frac{\tan^2 \beta + \tan^2 \gamma}{\tan^2 \varphi} \right) \\ \Delta &\stackrel{\text{def}}{=} \frac{4}{3} \tan \varphi \frac{C_L}{C_D} \frac{u_* d_{50}}{k_v U(z - z_{int})} \end{aligned} \quad (7.47)$$

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, [143]) for a fluid flow around a sphere:

$$\begin{aligned} C_D &= \begin{cases} \frac{24}{Re} + \frac{\frac{2.6Re}{5}}{1 + \left(\frac{Re}{5}\right)^{1.52}} + \frac{0.411 \left(\frac{Re}{263'000}\right)^{-7.94}}{1 + \left(\frac{Re}{263'000}\right)^{-8}} + \frac{Re^{0.8}}{461'000}, & 1.0 * 10^2 \leq Re \leq 1.0 * 10^6 \\ 1.0, & Re \leq 1.0 * 10^2 \\ 0.2, & 1.0 * 10^6 \leq Re \end{cases} \end{aligned} \quad (7.48)$$

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

$$Re \stackrel{\text{def}}{=} \frac{U_\infty d_{50}}{\nu} \quad (7.49)$$

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 assumed the form:

$$F_L = \frac{4}{3} \rho C_L \frac{\pi D^3}{8} U \frac{u_*}{k_v(z - z_{int})} \quad (7.50)$$

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, [177]):

$$C_L = \begin{cases} (9.0 * 10^{-5}) Re^{0.882}, & 2.0 * 10^3 \leq Re \leq 1.75 * 10^4 \\ 0.07, & Re \leq 2.0 * 10^3 \\ 0.5, & 1.75 * 10^4 \leq Re \end{cases} \quad (7.51)$$

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:

$$p_0 = p_b + (z_b - z_0) \gamma_m + \frac{1}{2} \rho U_{nor,b}^2, \quad U_{nor} = |(\underline{u}_b - \underline{u}_0) \cdot (\underline{x}_b - \underline{x}_0)| \quad (7.52)$$

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.

7.3. A simplified approach for soil liquefaction

This section refers to Amicarelli et al. (2016, [7]).

A simplified approach is implemented for soil liquefaction, by means of linearized pore pressure functions, depending on the critical number N of equivalent uniform cycles:

$$\frac{p_f(\underline{x}, t) - p_{f,nq}(\underline{x}, t)}{p_{nq}(\underline{x}, t) - p_{f,nq}(\underline{x}, t)} = f_{liq} \left(\frac{N(t)}{N_L} \right) \quad (7.53)$$

where the subscript “ nq ” represents “no-quake” conditions, N_L the critical value of N (when liquefaction occurs) and N/N_L is named cyclic number ratio.

In case the liquefaction time (t_{liq}) is known and f_{liq} is approximately linear, (7.53) becomes:

$$\sigma'_m(\underline{x}, t) = \begin{cases} \sigma'_{m,nq}(\underline{x}, t), & t \leq t_{q0} \\ \sigma'_{m,nq}(\underline{x}, t) \left[1 - \frac{(t - t_{q0})}{(t_{liq} - t_{q0})} \right], & t_{q0} < t \leq t_{liq} \\ 0, & t_{liq} < t \leq t_{qf} \\ \sigma'_{m,nq}(\underline{x}, t), & t > t_{qf} \end{cases} \quad (7.54)$$

where t_{q0} and t_{qf} are the quake starting and ending times.

8. THE DB-SPH BOUNDARY TREATMENT SCHEME

This section describes the “Discrete Boundary” (DB) - SPH method for boundary treatment (Amicarelli et al., 2013, [104]). 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: DB-SPH particle approximation and modifications of the balance equations (Sec.8.1); 1D Linearized Partial Riemann Solver (Sec. 8.2); semi-particle volume (Sec.8.3); DB-SPH inlet and outlet sections (Sec.8.4); shear stress boundary terms (Sec.8.5).

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:

$$\left\langle \frac{\partial f}{\partial x_i} \right\rangle_{x_0} = \sum_a f_a W_a n_{i,a} \omega_a - \sum_b f_b W'_b \omega_b, \quad W'_b \equiv \frac{\partial W}{\partial x_i} \Big|_b \quad (8.1)$$

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:

$$\left\langle \frac{\partial f}{\partial x_i} \right\rangle_{x_0} \equiv \sum_a (f_a) \frac{W_a}{\gamma_0} n_{i,a} \omega_a - \sum_b (f_b) \frac{W'_b}{\gamma_0} \omega_b, \quad \gamma \equiv \int_{V_h} W dx^3 \quad (8.2)$$

γ varies as function of the involved computational particle “ ϕ ”. 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, [57]) 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, [57]).

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 $\frac{dx_w}{2}$. 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, [57]):

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

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:

$$\tilde{\gamma} \equiv \begin{cases} \sigma, & \gamma \geq \sigma + \varepsilon \\ \gamma, & \gamma < \sigma + \varepsilon \end{cases}, \quad \sigma \equiv \sum_{bs} W_{bs} \omega_{bs} = \sum_{bs} W_{bs} \frac{m_{bs}}{\rho_{bs}} \quad (8.4)$$

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, [57]), as synthesized by (8.5) and (8.6). Consider the Lagrangian derivative of γ :

$$\frac{d\gamma}{dt} \equiv \sum_a W_a \underline{n}_a \cdot (\underline{u}_a - \underline{u}_0) \omega_a, \quad \frac{\partial W}{\partial t} = 0 \quad (8.5)$$

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:

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

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, [125]), 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 the computational particle (f alternatively refers to density and every velocity component):

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

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):

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

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:

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

8.3. Semi-particle volume

The volume of a semi-particle is defined in Amicarelli et al. (2013, [6]):

$$\omega_s = k_w k_d dx_w \omega_a \quad (8.10)$$

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:

$$k_d = \begin{cases} \frac{\sum_i \alpha_i}{D \frac{\pi}{2} n_{af}} - 1 = \frac{\sum_i \alpha_i}{\frac{\pi}{2} n_{af}} - 1, & \sum_i \alpha_i > \frac{\pi}{2} n_{af} \\ 0, & \sum_i \alpha_i \leq \frac{\pi}{2} n_{af} \end{cases} \quad (8.11)$$

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 $+90^\circ$. The reference formula for α_i reads:

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

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.

8.5. Shear stress boundary terms

This section refers to Amicarelli et al. (2016, [7]), whose reading is suggested for further details.

The shear stress boundary terms of Ferrand et al. (2013, [57]) are adapted to treat mobile frontiers and integrated in the boundary treatment scheme of Amicarelli et al. (2013, [6]). Such a scheme is potentially suitable to represent the bottom drag between saturated granular material and a base rock, even during an earthquake.

The boundary terms above appear in the Right Hand Side of the momentum equation and are due to both surface wall elements and semi-particles. In the first case, one obtains:

$$\begin{aligned} & -\frac{1}{\gamma_0 \rho_0} \sum_a |\nabla \gamma_{0a}| (\mu_0 \nabla \underline{u}_0 + \mu_a \nabla \underline{u}_a) \cdot \underline{n}_a = \\ & = -\frac{1}{\gamma_0 \rho_0} \sum_a W_a \omega_a (\mu_0 \nabla \underline{u}_0 + \mu_a \nabla \underline{u}_a) \cdot \underline{n}_a, \quad \nabla \gamma_{0a} = W_a \omega_a \underline{n}_a \end{aligned} \quad (8.13)$$

where γ is the integral Shepard coefficient and \underline{n} the normal to any wall surface (neighbouring) element “ a ” of area ω_a .

Considering the derivation chain rule, one obtains:

$$\begin{aligned} \frac{\partial u_i}{\partial x_k} n_k &= \frac{\partial u_i}{\partial x_k} \cos(\alpha_{nk}) = \frac{\partial u_i}{\partial x_k} \frac{\partial x_k}{\partial n_k} = \frac{\partial u_i}{\partial \underline{n}} \Rightarrow \\ \Rightarrow (\mu_0 \nabla \underline{u}_{i,0} + \mu_a \nabla \underline{u}_{i,a}) \cdot \underline{n}_a &= \left(\mu_0 \frac{\partial u_{i,0}}{\partial \underline{n}} + \mu_a \frac{\partial u_{i,a}}{\partial \underline{n}} \right) \end{aligned} \quad (8.14)$$

Once provided the similarity law of the viscous sub-layer, aligned the friction velocity vector with fluid velocity close to wall and assumed mobile frontiers, one can write:

$$\frac{\partial u_{i,0}}{\partial \underline{n}} = \frac{(u_{i,0} - u_{i,a})}{d_{0a}} \quad (8.15)$$

where d_0 is the distance between the computational particle and the surface wall element.

Assuming a continuous field for the shear stress, the last term of (8.15) can be expressed as a SPH approximation and normalized by the discrete Shepard coefficient (σ):

$$\mu_a \frac{\partial u_{i,a}}{\partial \underline{n}} = \frac{1}{\sigma_a} \sum_0 \mu_0 \frac{(u_{i,0} - u_{i,a})}{d_0} W_0 \omega_0 \quad (8.16)$$

Thus, the shear stress boundary terms due to the surface wall elements are:

$$-\frac{1}{\gamma_0 \rho_0} \sum_a W_a \omega_a \left[\mu_0 \frac{(u_{i,0} - u_{i,a})}{d_{0a}} + \frac{1}{\sigma_a} \sum_0 \mu_0 \frac{(u_{i,0} - u_{i,a})}{d_0} W_0 \omega_0 \right] \cdot \underline{n}_a \quad (8.17)$$

The analogous term due to the semi-particles “ s ” reads:

$$2v_s \sum_s \frac{m_s}{\rho_0 r_{0s}} (\underline{u}_s - \underline{u}_0) \frac{\partial W}{\partial r} \Big|_s, \quad v_s \equiv v_a = \frac{\sum_0 v_0 W_0 \omega_0}{\sum_0 W_0 \omega_0} \quad (8.18)$$

9. TIME INTEGRATION SCHEMES (LEAPFROG, EULER, HEUN)

Time integration is ruled by a second-order Leapfrog scheme (refer to [212] for stability analysis and time integration schemes in SPH modelling), as described in Amicarelli et al. (2015, [7]) and Di Monaco et al. (2011, [46]):

$$\begin{aligned} x_{i,0}|_{t+dt} &= x_{i,0}|_t + u_{i,0}|_{t+dt/2} dt, \quad i=1,2,3 \\ u_{i,0}|_{t+dt/2} &= u_{i,0}|_{t-dt/2} + \left\langle \frac{du_{i,0}}{dt} \right\rangle_t dt, \quad i=1,2,3 \\ \rho_0|_{t+dt} &= \rho_0|_t + \left\langle \frac{d\rho_0}{dt} \right\rangle_{t+dt/2} dt \end{aligned} \quad (9.1)$$

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:

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

The scheme above can be rearranged in the following form:

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

where the subscripts here represent the time step ID.

RK2 assumes the following form:

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

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).

Time integration is constrained by the following stability criteria:

$$dt = CFL * \min_0 \left\{ \frac{2h^2}{v_0}; \frac{2h}{c + |u_0|}; \frac{2h^2}{v_{M,0}} \right\} \quad (9.5)$$

where dt (s) is the time step duration and CFL the Courant-Friedrichs-Lewy number. Following [1], the viscous term stability parameter is set to $C_v=0.05$.

An a-priori estimation of the elapsed time (t_e) can assume the following form:

$$t_e \propto \frac{1}{dx^{D+A}} \propto N^{\frac{D+A}{3}} \Rightarrow \frac{t_{e1}}{t_{e2}} = \left(\frac{dx_2}{dx_1} \right)^{D+A} \quad (9.6)$$

where D is the domain dimensionality and A varies from 1 to 2.

Under the simplifying hypothesis that the particle system of equations are independent, one obtains (at a fixed time step duration):

$$t_e \propto \frac{1}{dx^D} \Big|_{dt=\overline{dt}} \propto N \quad (9.7)$$

In case all the time steps are ruled by the *CFL* criterion, then $A=1$ and the following relationships are valid (at a fixed number of particles):

$$dt \propto h \propto dx \text{ thus } t_e \propto \frac{1}{dx} \Big|_{N=\overline{N}} \quad (9.8)$$

In case all the time steps are ruled by the stability criterion on the viscous term, then $A=2$ and the following relationships are valid (at a fixed number of particles):

$$dt \propto h^2 \propto dx^2 \text{ thus } t_e \propto \frac{1}{dx^2} \Big|_{N=\overline{N}} \quad (9.9)$$

In order to understand which stability criterion dominates, one considers the following ratio:

$$\frac{dt_{C_v}}{dt_{CFL}} = \frac{C_v h (c + |u|_{\max})}{CFL \cdot v_{\max}} \quad (9.10)$$

Provided a time step, if the ratio (9.10) is greater than 1, then the *CFL* criterion dominates. Otherwise, the viscosity stability criterion dominates.

When the *CFL* criterion dominates, the maximum viscosity has never any effect on dt if this condition is respected:

$$v_{\max} \leq \frac{C_v h c}{CFL} \Rightarrow dt \neq f(v_{\max}) \text{ anytime} \quad (9.11)$$

The elapsed time also depends on the global specific surface of the fluid domain. The highest the latter, the lowest the first. The global specific surface of the fluid domain is related to the mean number of neighbouring particles.

10. THE SUBSTATION-FLOODING DAMAGE SCHEME

A substation-flooding damage model is presented and integrated in SPHERA (Amicarelli et al., 2018, [3]). The model distinguishes the damage due to power outages from the damage to the components of the electrical substations.

Considering the power outages (blackout events), a vulnerability and “proxy” damage (in time units, not in monetary units) model is presented. This assumes, as a simplifying hypothesis of no redundancy of the electric grid, that the failure of an electrical substation triggers a blackout event in a grid branch (no matter about its length and connections). The assessment of the overall vulnerability of the electric grid and the global damage (in monetary units) due to flood-related blackout events is out of the targets of this code version and needs the following elements: the coupling of the present model with a power grid model; the maps of the exposed population, the values of the public (also environmental) and private goods, the activities affected by the blackout and their flood-related vulnerability curves. On the other hand, considering the components of the electrical substations, a complete (direct and tangible) damage model is presented.

The substation-flooding damage model estimates the following physical quantities, at every electrical substation: the Probability of a power Outage Start (*POS*) as function of the maximum substation water depth; the Expected power Outage Status (*EOS*), at the level of the single electrical substations (in the absence of power grid redundancy); Expected Outage Time/duration (*EOT*, s); the flood-related vulnerability and damage (euros) limited to the components of the electrical substations.

10.1. Mathematical models

The mathematical models of the substation-flooding damage scheme separately considers both blackout events (Sec.10.1.1) and the components of the electrical substations (Sec.10.1.2).

10.1.1. Proxy damage and vulnerability to flood-induced blackout events, in the absence of redundancy (at the level of the single electrical substations)

A proxy damage quantifies in time units (not in monetary units) the damage due to blackout events triggered by substation flooding. One assumes that the malfunction of a substation determines a blackout event in a branch of the power grid (simplifying hypothesis of no redundancy).

The breakdown in the electrical energy supply is analysed only considering the power outage events and neglecting the damage due to brownouts (voltage drops).

The Probability of an Outage Start (*POS*) (i.e. the probability of triggering a blackout event) at a fixed time is smaller than (or equal to) the probability of occurrence of a blackout event at the same time (*EOS*, “Expected Outage Status”) because the latter also depends on the blackout events which might start previously.

The procedure of HAZUS-MH (2011, [86]) considers a threshold water depth of $Y_{th}=1.2\text{m}$ for blackout events associated with the flooding of electrical substation. However, the above procedure does not mention neither data nor the method to elaborate them. Holmes (2015, [88]) explains that the above threshold value is overestimated. In particular, the majority of the UK electrical substations has shown a threshold of $Y_{th}=0.30\text{m}$ (Crawford & Seidel, 2013,[39]). Considering this value and a maximum threshold of $Y_{th,max}=0.50\text{m}$, Holmes (2015, [88]) provides a linear relationship between the probability of an Outage Start and water depth, here assumed as spatial average over the territory belonging to the electrical substation (Y_{sub} , m):

$$POS|_{t=t^*} = \begin{cases} 1, & Y_{sub}|_{t=t^*} \geq 0.52 \\ 1.92Y, & 0 < Y_{sub}|_{t=t^*} < 0.52 \\ 0, & Y_{sub}|_{t=t^*} = 0 \end{cases} \quad (10.1)$$

The above relationship has been validated on other open-field data (Holmes, 2015, [88]).

The “Expected Outage Status” is defined as a binary variable which represents either the expected presence ($EOS=1$) or the expected absence ($EOS=0$) of a blackout event at a fixed time, at the level of the single electrical substation:

$$EOS|_{t=t^*} = \begin{cases} 1, & \sum_i \max\left\{\left(POS|_{t=t_i} - 0.49\right), 0\right\} > 0, \quad t^* - t_{rei,e} \leq t_i \leq t^* \\ 0, & otherwise \end{cases} \quad (10.2)$$

EOS is unity if at least once, during the period before the on-going time ($t=t^*$) lasting the restoration time of the electrical infrastructures ($t_{rei,e}$), POS is greater than 0.5.

The values of $t_{rei,e}$ are commonly greater for smaller substations with less redundancy. This characteristic time has been quantified by several authors: Chow et al. (1996, [35]) report the interval $t_{rei}=0'-500'$; Maliszewski & Perrings (2012, [122]) suggest an average value of $t_{rei,e}=99'$ and a maximum value of $t_{rei,max}=330'$. Both the above references consider a typical value of t_{rei} smaller than two hours. However, their analyses only consider blackout events under normal (non-extreme) environmental conditions. Reed (2008, [168]) proposes to use a particular realization of the “gamma” probability density function, for t_{rei} on blackouts induced by intense meteorological and flood events.

Considering a null minimum value and a maximum value (under the worst hypothesis) of 22 hours (95th percentile of the “gamma” distribution of Reed -2008, [168]-), the present model assumes an expected restoration time of 11 hours ($t_{rei,e}=39'600s$), in the absence of further data and in favour of safety.

The system (10.1)-(10.2) represents a formulation of the vulnerability of the single electrical substation to the flood-induced blackout events, in the absence of redundancy.

The “Expected Outage Time” (EOT , s) is here defined as the expected cumulated duration of the sub-periods of blackout:

$$EOT|_{t=t^*} = \sum_i EOS_{t=t_i} \Delta t|_{t=t_i}, \quad t_0 \leq t_i \leq t_f \quad (10.3)$$

where Δt is the model time step duration. The subscripts “ 0 ”, “ f ” and “ i ” represent initial conditions, final conditions and a generic time step, respectively. A blackout event can involve several interruptions and the following relationship is assumed: $EOT \geq t_{rei,e}$. Eq.(10.3) represents a formulation of the “proxy” damage (quantified in time units) due to flood-induced blackout events, in the absence of redundancy(at the level of the single substations).

10.1.2. Flood-induced damage to the components of the electrical substations

This mathematical model assesses the (direct and tangible) damage induced by floods to the components of the electrical substations (D_{sub}). This damage model is complete as it quantifies both vulnerability and damage (the latter in monetary units):

$$D_{sub}|_{t=t^*} = V_{u,sub}|_{t=t^*} \cdot V_{a,sub} \quad (10.4)$$

The value of the electrical substations ($V_{a,sub}$) can be expressed in US dollars (HAZUS-MH, 2011, [86]):

$$V_{a,sub} = \begin{cases} 50'000, & High - Voltage - Substation \\ 20'000, & Medium - Voltage - Substation \\ 10'000, & Low - Voltage - Substation \end{cases} \quad (10.5)$$

but SPHERA works in euros (currency exchange rate of February 2018):

$$V_{a,sub} = \begin{cases} 41'000, & \text{High-Voltage-Substation} \\ 16'400, & \text{Medium-Voltage-Substation} \\ 8'200, & \text{Low-Voltage-Substation} \end{cases} \quad (10.6)$$

$V_{u,sub}$ is the vulnerability of an electrical substation with respect to the (direct and tangible) flood-induced damage involving the substation components. This model provides a regression curve (6th degree polynomial function) for $V_{u,sub}$, after elaboration of the point values reported by HAZUS-MH (2011, [86]):

$$V_{u,sub}|_{t=t^*} = \begin{cases} 0.15, & (Y_{sub,max}|_{t \leq t^*}) \geq 10 \\ \left(a(Y_{sub,max}|_{t \leq t^*})^6 + b(Y_{sub,max}|_{t \leq t^*})^5 + c(Y_{sub,max}|_{t \leq t^*})^4 + \right. \\ \left. + d(Y_{sub,max}|_{t \leq t^*})^3 + e(Y_{sub,max}|_{t \leq t^*})^2 + f(Y_{sub,max}|_{t \leq t^*}) \right), & 0 < (Y_{sub,max}|_{t \leq t^*}) < 10 \\ 0, & (Y_{sub,max}|_{t \leq t^*}) = 0 \end{cases} \quad (10.7)$$

where the maximum value of the substation water depth $Y_{sub,max}$ refers to the period simulated until the on-going time ($t=t^*$). The regression procedure provides the following constants: $a=-1.22877 \cdot 10^{-6} \text{m}^{-6}$, $b=1.92478 \cdot 10^{-5} \text{m}^{-5}$, $c=8.4216 \cdot 10^{-6} \text{m}^{-4}$, $d=-0.00119121 \text{m}^{-3}$, $e=0.00390726 \text{m}^{-2}$, $f=0.0170243 \text{m}^{-1}$.

10.2. The numerical models

The numerical models of the substation-flooding damage scheme discretize the mathematical models of Sec.10.1. At the beginning of the simulation the following procedures are executed:

- ✓ identification of the DEM vertices internal to the polygons (representing the electrical substations in plan view);
- ✓ assessment of the areas of the above polygons (represented by triangles, quadrilaterals, pentagons and hexagons);
- ✓ initializations of the variables of the electrical substations.

At the end of each output writing time step of the damage model, the following procedures are executed for every electrical substation:

- ✓ assessment of the variable Y_{sub} as spatial average of the water depth values at the DEM vertices within the polygon of the electrical substation;
- ✓ update of the variable $Y_{sub,max}$;
- ✓ assessment of the variables POS and EOS ;
- ✓ update of the variables EOT , D_{sub} and $V_{u,sub}$;
- ✓ writing of the output file for the vulnerability and damage variables.

The water depths at the DEM points are saved and stored until the following time step.

As an example, the map of the Italian electrical stations, substations and transmission lines is available at MATTM (2018, [133]). It is useful to activate the layer of OpenStreetMap and the “aerial view” of Bing to detect the electrical stations (squares), substations (polygons) and transmission lines (lines), as well as the electricity pylons (nodes/points). Contrarily, the feature “Atlatete Linee” seems more imprecise (some symbols of the electrical stations would represent electricity pylons; the electrical substations do not seem to represent the nodes of the transmission power grid).

11. THE MODELLING CHAIN

An overview of the numerical modelling chain of SPHERA is reported in Figure 11.1.

Published at the end of 2014, the dataset SRTM3 (USGS, [192]) provides a repository of DEM (“Digital Elevation Models”) with an almost global cover and spatial resolution of 1” (in terms of geographical coordinates) or ca.31m (maximum/coarser spatial resolution in terms of cartographic coordinates). With respect to the former free DEM datasets (finest spatial resolution of 3”), SRTM3 has permitted a relevant improvement in using Open-Access DEM. The SRTM3 products are available in the format “.tif”. Considering its global coverage, SRTM3 provides a root-mean-square error on heights of ca.6m, even though the error kurtosis is very high ([170]). However, these estimations refer to almost the whole terrestrial surface, included the high-latitude regions where errors are definitely higher.

The software tool GDAL (OSGEO, [65]), the main QGIS library, can be used as an independent code. In the frame of the present modelling chain, GDAL allows to convert the DEM file format “.tif” in the alternative format “.dem”.

DEM2xyz (RSE SpA, [39]) reads the DEM file (“.dem” format), converts the geographic coordinates in Cartesian coordinates over a regular grid and writes the resulting DEM on an output file (“.xyz” format), possibly coarsening the spatial resolution.

Paraview (Kitware, [153]) reads the “.xyz” output file of DEM2xyz and elaborates a 2D Delaunay grid starting from the DEM vertices. Paraview also allows to cut the numerical domain (the cuts have to be far enough from the water bodies not to disable the procedures to extrude the water bodies from the DEM), circumscribes the water bodies, draws the possible filing/digging regions, detects the dam toe and the most upstream point over the coastline of the water bodies.

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

The numerical tool ply2SPHERA_perimeter (RSE SpA, [160]) converts the DEM “.ply” file in two distinct output files. They have the same format as the sections “VERTICES” and “FACES” of the main input file of SPHERA. It is the vertices and faces of the portion of the DEM within the numerical domain of SPHERA.

In case the boundary treatment method of Sec.8 is used, SnappyHexMesh (OpenFOAM, OpenCFD Ltd, [153]) is used as a surface grid generator for the initial positioning grid of the DB-SPH elements.

Once the sections “VERTICES” and “FACES” are obtained from ply2SPHERA_perimeter and the input file for the positioning surface grid is produced by SnappyHexMesh, one completes the remaining sections of the main input file of SPHERA.

This 3D CFD-SPH code is executed to simulate the propagation of floods in non-stationary regime with transport of granular material and solid bodies (the other application fields the code has been applied to are: fast landslides, sea waves; sloshing fuel tanks; sediment removal from water reservoirs).

The output files of SPHERA which contain the profiles (1D) of the fluid dynamics variables are visualized by means of Gnuplot (Williams & Kelley, [69]), which returns the output file in the “.eps” format. These files are read by GSView (Ghostgum Software Pty Ltd, [79]) and converted in the “.png” format. The profiles simulated are compared with the analogous experimental (or numerical) profiles available from experimental images or the scientific literature (indexed journals; Open-Data archives). In these cases it is normally admitted the digitization of experimental

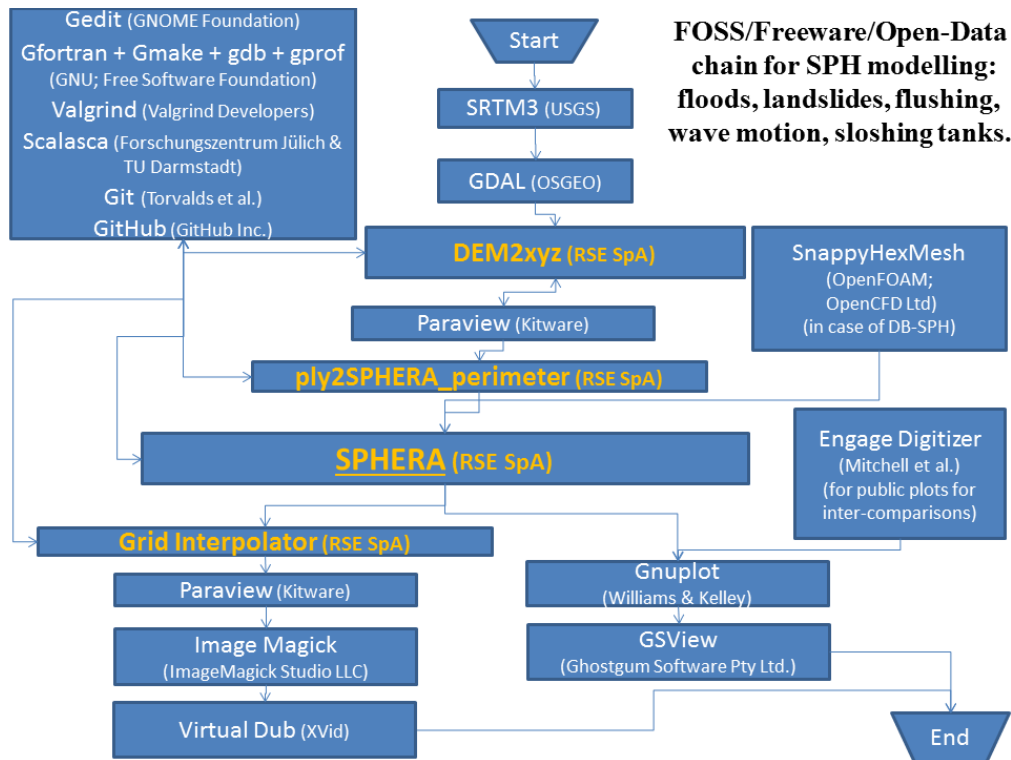


Figure 11.1. FOSS / Freeware / Open-Data tools of the modelling chain for SPHERA.

and numerical profiles from published sources by means of Engage Digitizer (Mitchell et al., [48]), with proper citation of the source, in order to validate the code.

The output files of SPHERA which contains the synthetic fluid dynamics fields need a following elaboration by means of Grid Interpolator (RSE SpA, [79]).

Paraview shows the (2D and 3D) fluid dynamics fields produced by SPHERA and returns the associated image files. These are concatenated in “.gif” animations by means of Image Magick (ImageMagick Studio LLC, [90]). The compression of these animations, necessary in case the concatenation involves many files, is carried out by means of Virtual Dub ([213]), which returns an “.avi” video output file.

The RSE tools of the present numerical chain (i.e. SPHERA, DEM2xyz, ply2SPHERA_perimeter, Grid Interpolator) are developed by means of a series of numerical tools: gedit (GNOME Foundation, [66], text editor), gfortran (GNU, Free Software Foundation, [69]; Fortran compiler), gmake (GNU, Free Software Foundation, [72]; for the Makefile execution), gdb (GNU, Free Software Foundation, [66]; debugger), gprof (GNU, Free Software Foundation, [76]; profiler for scalar executions), Valgrind (Valgrind Developers, [204]; memory management tool), Scalasca (Forschungszentrum Jülich & TU Darmstadt, [173]; code profiler for OMP executions). The RSE tools are developed and distributed with no charge by means of Git (Torvalds et al., [69]; main “Distributed Version Control System” -DVCS-) and GitHub (GitHub Inc, [70]; main platform for Git-managed software).

The numerical modelling chain is based on free tools: FOSS, freeware or OpenData.

FOSS tools are defined as “Free/Libre and Open-Source Software” by the Free Software Foundation ([62]). The FOSS tools of the modelling chain are: gedit, gfortran, gmake, gdb, gprof, Valgrind, Scalasca, Git, GDAL, Paraview, DEM2xyz, SnappyHexMesh, ply2SPHERA_perimeter, SPHERA, Grid Interpolator, Image Magick, Gnuplot, Engage Digitizer, Virtual Dub.

“Freeware” tools are simply free software. Two freeware tools are used in the reference modelling chain: GitHub (for public repositories) and GSView.

“Open-Data” tools are databases available upon public and free access like SRTM3, which belongs to the modelling chain.

The FOSS tools gfortran, gprof and Scalasca can be possibly replaced with more effective codes such as: ifort (Intel Corporation, [91]; Fortran compiler); idb (Intel Corporation, [91]; Fortran debugger); cpuinfo (Intel Corporation, [91]; code profiler for OMP executions); ITAC, Trace Analyzer and VTUNE (Intel Corporation, [91]; code profiler for OMP/MPI executions); Advisor (Intel Corporation, [91]; profiler for executions with code vectorization); TotalView (Rogue Wave Software, [199]; code debugger for parallel executions). Analogously, it is possible to support or replace the other elements of the modelling chain with more effective software, if available (e.g., a DEM with finer spatial resolution).

The replacement of a free tool with a proprietary tool (available with charge) is normally a reversible procedure which does not alter the functioning of the modelling chain.

12. DEVELOPER GUIDE

The developer guide is represented by the following sub-sections: a synthetic description of the program units (Sec.12.1); the style formatting (Sec.12.2); the modifications with respect to SPHERA v.8.0 (Sec.12.3).

12.1. SPHERA v.9.0.0: synthetic description of the program units

The following sub-sections briefly describe all the program units of SPHERA v.9.0.0, according to their reference folder.

12.1.1. 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 12.1).

12.1.2. 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 12.2).

12.1.3. 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 12.3).

12.1.4. Program units for the transport of solid bodies

The folder “Body_Transport” contains the program units exclusively dedicated to the transport of solid bodies (Amicarelli et al., 2015, [7]; Table 12.4).

12.1.5. Program units for the constitutive equation

The folder “Constitutive_Equation” contains the program units for the constitutive equation (Table 12.5; Amicarelli et al., 2017, IJCFD).

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).
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 12.1. Program units for the boundary conditions of the inlet/outlet sections (“BC”; SPHERA v.9.0.0).

Program unit	Synthetic Description
CalcPre	Particle pressure estimation.
Continuity_Equation	To accumulate contributions for the 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 12.2. Program units for the continuity equation (“BE_mass”; SPHERA v.9.0.0).

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).
velocity_smoothing	To calculate a corrective term for velocity.
velocity_smoothing_SA_SPH_2D	To calculate a corrective term for velocity.
velocity_smoothing_SA_SPH_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 12.3. Program units for the momentum equation (“BE_momentum”; SPHERA v.9.0.0).

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 12.4. Program units for the transport of solid bodies in free surface flows (“Body_Transport”; SPHERA v.9.0.0).

Program unit	Synthetic Description
mixture_viscosity	To compute the frictional viscosity and the mixture viscosity for dense granular flows (KTGF packing limit). (Amicarelli et al., 2017, IJCFD)
Viscapp	Constitutive equation with tuning parameters (validated in Manenti et al., 2012, JHE)

Table 12.5. Program units for the constitutive equation (“Constitutive_Equation”; SPHERA v.9.0.0).

12.1.6. 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, [6]; Table 12.6).

12.1.7. 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, [122]) and its further developments (Table 12.7).

12.1.8. 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 12.8).

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_BC_shear_viscosity_term	Computation of the contributions to the numerator of the boundary shear viscosity term in DB-SPH-NS.
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).
DBSPH_velocity_gradients_VSL_SNBL	Computation of the velocity gradients in the Viscous Sub-Layer of the Surface Neutral Boundary Layer. The gradients are used in the DB-SPH BC shear viscosity term (DB-SPH-NS). For wall elements, the numerator and the denominator (wall element Shepard coefficient without contributions from semi-particles) are updated independently. Their ratio is computed in "DBSPH_BC_shear_viscosity_term": here are summed their contributions. To compute the kinematic viscosity of the semi-particles (before Shepard correction). Contributions to the discrete Shepard coefficient of wall elements depending on fluid particles (not on semi-particles).
drafts	Wall element contributions for Monaghan artificial viscosity term.
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.
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 12.6. Program units for the boundary treatment scheme DB-SPH (“DB_SPH”; SPHERA v.9.0.0).

Program unit	Synthetic Description
compute_k_BetaGamma	To compute $k_BetaGamma = \frac{\eta}{\rho c}$. $k_BetaGamma$ is the ratio between

drafts	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	Mohr-Coulomb 2D erosion criterion (Manenti et al., 2012, JHE). Shields erosion criterion works better (Manenti et al., 2012, JHE).
Shields	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). 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 12.7. Program units for the erosion criterion (“Erosion_Criterion”; SPHERA v.9.0.0).

Program unit	Synthetic Description
area_hexagon	Computation of the area of a generic hexagon from the coordinates of its vertices.
area_pentagon	Computation of the area of a generic pentagon from the coordinates of its vertices.
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_convex_non_degenerate_polygon	Test to evaluate if a point lies inside or strictly outside a polygon. A point is internal to the polygon if its distances from the lines passing for the polygon sides (no matter about the number of sides, but they must be taken in either a clockwise or an anti-clockwise order), have all the same sign of a generic polygon vertex not belonging to the selected side -a null distance is always a positive test for internal points-). The maximum number of polygon sides is now equal to 6 (triangles, quadrilaterals, pentagons and hexagons can be treated). Polygons must be convex and non-degenerate (a n-side polygon should have n vertices, not more).
point_inout_hexagon	Test to evaluate if a point lies inside or strictly outside a generic hexagon. The hexagon is partitioned into 4 triangles (P1P2P6,P2P5P6,P2P3P5,P3P4P5). A point is internal to the hexagon if it is internal to one of its triangles.
point_inout_pentagon	Test to evaluate if a point lies inside or strictly outside a generic

point_inout_quadrilateral	pentagon. The pentagon is partitioned into 3 triangles (P1P2P5,P2P3P5,P3P4P5). A point is internal to the pentagon if it is internal to one of its triangles. Test to evaluate if a point lies inside or strictly outside a generic quadrilateral. The quadrilateral is partitioned into 2 triangles (P1P2P3,P1P3P4). A point is internal to the quadrilateral if it is internal to one of the triangles.
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_axis_angle	Provided 2 vectors, this subroutine computes the rotation axis and the rotation angle which allow rotating from the unit vector aligned with the first vector to the unit vector aligned with the second vector.
vector_rotation_Euler_angles	3D rotation of a given vector, provided the vector of Euler's angles (3D).
vector_rotation_Rodrigues	3D rotation of a given vector, provided the rotation axis and the rotation angle, based on Rodrigues formula.

Table 12.8. Program units on Geometry (“Geometry”; SPHERA v.9.0.0).

12.1.9. Program units for the initial conditions (IC)

The folder “IC” contains the program units on the management on the initial conditions (Table 12.9).

12.1.10. 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”.

12.1.11. 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 12.10).

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).
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.
SetParticleParameters	Setting initial particle parameters.
SetParticles	Particle coordinates (initial conditions).
SubCalcPreIdro	Hydrostatic pressure profiles (in case they are imposed as initial conditions).

Table 12.9. Program units for the initial conditions (“IC”; SPHERA v.9.0.0).

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 12.10. Program units for the main algorithms (“Main_algorithm”; SPHERA v.9.0.0).

12.1.12. Modules

The folder “Modules” contains the Fortran modules of SPHERA v.9.0.0. (Table 12.11).

12.1.13. 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 12.12).

12.1.14. Program units for post-processing

The folder “Post_processing” contains the program units to post-process the code results (Table 12.13). 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”).

12.1.15. Program units for pre-processing

The folder “Pre_processing” contains the program units (Table 12.14) to pre-process the input files of SPHERA, which are:

- main input file (“.inp” format is defined in SPHERA v.9.0.0; 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_file_module	Module for I/O.
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 12.11. Fortran modules (“Modules”; SPHERA v.9.0.0).

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.
w	kernel function

Table 12.12. Program units for the neighbouring search, the smoothing operators and the interface detection (“Neighbouring_Search”; SPHERA v.9.0.0).

Program unit	Synthetic Description
calc_pelo	Post-processing to write the free surface height.
CalcVarp	To calculate physical quantities at a monitoring point.
cat_post_proc	To concatenate the ".txt" output files and remove the original ones.
CreateSectionPoints	Minor program unit
electrical_substations	output assessment and writing for electrical substations (only in 3D). Output (time series): Probability of an Outage Start (POS), Expected Outage Status (EOS), Expected Outage Time (EOT, time series update of its expected scalar value), Damage to the Substation Beyond Outage (Dsub, time series update of its expected scalar value), Substation Vulnerability (Vul). Output depends on Ysub (spatial average of the fluid/mixture depth at the DEM grid points, within the substation polygon).
GetVarPart	Getting particle values.
interface_post_processing	Post-processing the interfaces for bed-load transport phenomena.
Memo_Ctl	Post-processing for monitoring lines and points.
Memo_Results	To write detailed results for restart. 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	To print 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 12.13. Program units for post-processing (“Post_processing”; SPHERA v.9.0.0).

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 transport 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 12.14. Program units for pre-processing (“Pre_processing”; SPHERA v.9.0.0).

Program unit	Synthetic Description
AddBoundaryContribution_to_CE2D	To compute boundary terms for the 2D continuity equation (rodivV). Equation refers to particle np_i. 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 np_i. 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 np_i. (Di Monaco et al., 2011, EACFM).
AddBoundaryContributions_to_ME3D	To compute boundary terms for 3D momentum equation (gradPsuro, ViscoF). Equations refer to particle np_i. It performs implicit computation of gradPsuro. (Di Monaco et al., 2011, EACFM).
AddElasticBoundaryReaction_2D	To add supplementarily 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 $R = (c_0^2/d) \ln(z_i/d)$ [for $z_i < d$], stemming from the compressible reaction of the fluid, where: $c_0^2 = E/\rho_0$ is the square of the sound speed within the fluid;

	<p>z_i is the distance of the particle P_i 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).</p>
AddElasticBoundaryReaction_3D	<p>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 $R=(c_0^2/d) \ln(z_i/d)$ [for $z_i < d$], stemming from the compressible reaction of the fluid, where: $c_0^2 = E/\rho_0$ is the square of the sound speed within the fluid; z_i is the distance of the particle P_i 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).</p>
BoundaryMassForceMatrix2D	Generation of the generalised boundary mass force matrix R_N , on the base of the cosine matrix T and the parameter F_i . (Di Monaco et al., 2011, EACFM)
BoundaryMassForceMatrix3D	Generation of the generalised boundary mass force matrix R_N , on the base of the cosine matrix T and the parameter F_i . (Di Monaco et al., 2011, EACFM)
BoundaryPressureGradientMatrix3D	To generate the pressure gradient matrix RRP , based on the cosine matrix T and the parameter vector Ψ_i . (Di Monaco et al., 2011, EACFM)
BoundaryReflectionMatrix2D	Generation of the generalised reflection matrix R , based on the cosine matrix T and the parameters Ψ_iS and Ψ_iN . (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	<p>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 $\text{BoundaryIntegralTab}()$ the coordinates (x,y,z) of each point are stored; in the forth column the relative d_α (portion of solid angle relative to the point, necessary for integrations) is stored. $\text{BITcols} = 4$. (Di Monaco et al., 2011, EACFM)</p>
ComputeBoundaryVolumeIntegrals_P0	(Di Monaco et al., 2011, EACFM)
ComputeKernelTable	<p>To pre-compute and store in $\text{kerneltab}(0:\text{ktrows}, 0:\text{kcols})$ the following values:</p> <p>$\text{kerneltab}(0:\text{ktrows}, 0) = \text{rob} = \text{rb}/h$ $\text{kerneltab}(0:\text{ktrows}, 1) = \text{Int } W * \text{ro}^2 \text{ dro}$ (from rob to 2) $\text{kerneltab}(0:\text{ktrows}, 2) = \text{Int } dW*/\text{dro ro dro}$ (from rob to 2) $\text{kerneltab}(0:\text{ktrows}, 3) = \text{Int } dW*/\text{dro ro}^2 \text{ dro}$ (from rob to 2) $\text{kerneltab}(0:\text{ktrows}, 4) = \text{Int } dW*/\text{dro ro}^3 \text{ dro}$ (from rob to 2) (Di Monaco et al., 2011, EACFM)</p>
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 $x_{pi}=\text{LocXY}(1,\text{icbs})$ and $y_{pi}=\text{LocXY}(2,\text{icbs})$, on the adjacent boundary side icbs . (Di Monaco et al., 2011, EACFM)
ComputeVolumeIntegral_WdV2D	Computing the integral of WdV extended to the volume delimited by the kernel support (radius= $2h$) of the particle i , whose local coordinates are $x_{pi}=\text{LocXY}(1,\text{icbs})$ and $y_{pi}=\text{LocXY}(2,\text{icbs})$, 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(isode)%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 np_i 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 np_i 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 np_i, and cancels the repeated ones. (Di Monaco et al., 2011, EACFM)
FindCloseBoundarySides2D	To finds the "close" boundary sides, i.e. those sided at a distance from particle np_i <= 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 np_i 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)

Intbcsides(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 12.15. Program units for the boundary treatment scheme SA-SPH (“SA_SPH”, SPHERA v.9.0.0)

Program unit	Synthetic Description
Euler	Explicit RK1 time integration scheme (Euler scheme).
Heun	Heun scheme: explicit RK2 time integration scheme.
time_step_duration	Computation of the time step duration (dt) according to stability constraints (CFL condition, viscosity term stability criterion, interface diffusion criterion - not recommended-). Plus, a special treatment for Monaghan artificial viscosity term and management of low-velocity SPH mixture particles for bed-load transport phenomena.
stoptime	Stopping time.
time_integration	Explicit Runge-Kutta time integration schemes
time_integration_body_dynamics	Euler time integration for body transport in fluid flows.

Table 12.16. Program units for time integration (“Time_integration”; SPHERA v.9.0.0).

12.1.16. *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, [46]; Table 12.15).

12.1.17. *Program units for managing Fortran character variables*

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

12.1.18. *Program units for time integration*

The folder “Time_integration” contains those program units, which concern RK1 and RK2 time integration schemes (Table 12.16).

12.2. *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.

12.3. Modifications with respect to SPHERA v.8.0

Hereafter is reported the list of modifications of SPHERA v.9.0.0, with respect to SPHERA v.8.0, according to git format.

commit de264ca645593d9a4c3a6acc41b7c204e4d088b8

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Fri Aug 3 12:19:36 2018 +0200

Minor modifications.

commit 75f50318377f954661da8de607e7e22693215902

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Fri Aug 3 10:33:56 2018 +0200

Minor modifications.

Folders "bin", "debug", "debug_omp": executable files of SPHERA v.9.0.0 are available.

commit 8f786fe307de381ddfcf76d1fecbecbc73afe182

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Wed Aug 1 12:13:29 2018 +0200

Minor modifications. Input. Modifications on the following test cases:

"db_squat_obstacle.inp"; "db_body_exp_UniBas.inp".

commit b6bbfc2c18b584981cfa3f6448f520fdcef58c61

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Wed Jul 25 10:04:45 2018 +0200

Modifications in progress for the release version SPHERA v.9.0.0.

Input. New and updated tutorials for SPHERA v.9.0.0.

Postprocessing. "*wall_IDs*" files and "*wall_regions*" files are concatenated (subroutine "cat_post_proc").

Pre-processing. Bug fixed on 2D restart: arrays "GCBFVector" and "GCBFPointers" only work in 3D.

commit abf4371b9e3032f53b7a143a5f3d7ebfaf82ac1b
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Wed Apr 4 16:07:56 2018 +0200

SPHERA v.9.0.0.

Version update in the program unit labels. Version string update for the input file check ("static_allocation_module.f90"). The Makefile is ready for a free optimized gfortran execution. Update of the description on the major numerical developments of the code.

commit 3e5b2fa8c5f4aef0fd6277d4565767a1ec845f4e
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Tue Apr 3 11:33:50 2018 +0200

Minor modifications.

Input. Tutorial: "Alpe_Gera_dbf_Lanzada_substations" .

commit b3e92d3ab826955c61f187974950463c5697e14c
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Wed Mar 14 08:51:55 2018 +0100

Minor modifications.

Documentation. Appendix update. Bulk modulus assignment.

commit 61ba5309eb85e2157ba8a2d1a768fbe800e96fac
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Mon Mar 5 16:00:50 2018 +0100

Minor modifications.

Input. New tutorials: "SPH_udb_Kim2015JH_scenario1_dx_0_025m_gate.inp",
"SPH_udb_Kim2015JH_scenario1_dx_0_025m_no_gate.inp",
"SPH_udb_Kim2015JH_scenario1_dx_0_050m_gate.inp",
"SPH_udb_Kim2015JH_scenario1_dx_0_050m_no_gate.inp",
"SPH_udb_Kim2015JH_scenario2_dx_0_025m_gate.inp",
"SPH_udb_Kim2015JH_scenario2_dx_0_025m_no_gate.inp",
"SPH_udb_Kim2015JH_scenario2_dx_0_050m_gate.inp",
"SPH_udb_Kim2015JH_scenario2_dx_0_050m_no_gate.inp".

commit 97320bc16e12849638ed13405efbab7c7d1ec427
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Mon Feb 26 17:29:14 2018 +0100

Minor modifications.

Post_processing. Fix on the subroutine "electrical_substations.f90" for restarted simulations.

commit 646d653b0b4edb8fbb0dd42d7941fd1770e2dfa1
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Thu Feb 22 14:59:48 2018 +0100

Minor modifications. Post_processing. Fix on the formula for vulnerability in

the program unit "electrical_substations.f90".

commit a71647aea6ca74cfb758364678db78e622d8c659

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Wed Feb 21 17:35:05 2018 +0100

Minor modifications.

commit bc6339cfb892d6536d86e87e9b1ffa56475731b9

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Mon Feb 19 12:00:10 2018 +0100

Minor modifications.

Post_processing. Subroutine "cat_post_proc.f90": fix on substations.

commit 6b21ef95d1dbfb14156909a629d5772d07617460

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Fri Feb 16 18:09:04 2018 +0100

Minor modifications.

Post_processing. Subroutine "cat_post_proc.f90": fix on substations.

Post_processing. Subroutine "electrical_substations.f90": iterative allocation and deallocation checks are not written anymore.

commit 31a8cb6398a413a8654a9396934cb093c8dc4d30

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Fri Feb 16 13:19:35 2018 +0100

Minor modifications.

Pre_processing. A generic vertex of any electrical susbtation only needs 2 coordinates.

commit 18b5258f2a5b4fa39d9a2b5c46ac6bfe109aeb06

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Thu Feb 15 16:27:15 2018 +0100

Minor modifications.

Geometry. New program units (after compilation): "area_hexagon", "area_pentagon".

Post_processing. New functionality to monitor electrical substations. New program unit (after compilation): "electrical_substations".

Post_processing. Z_fluid_step now belongs to the module "Dynamic_allocation_module" as "Z_fluid_max".

Pre_processing. New program unit (after compilation): "ReadSubstations".

Input. New/updated tutorials: "Alpe_Gera_dam_break_no_volume_correction.inp",

"Alpe_Gera_dam_break_no_volume_correction_flood_control_dam.inp",

"Alpe_Gera_dam_break_volume_correction.inp",

"Alpe_Gera_dam_break_volume_correction_flood_control_dam.inp",

"Alpe_Gera_double_dam_break_no_volume_correction.inp",

"Alpe_Gera_double_dam_break_volume_correction.inp",

"ICOLD_dam_breach_trunks.inp", "dam_breach_ICOLD_DB-SPH_BD.inp",

"edb_2D_FraCap02.inp", "edb_2D_FraCap02_Taipei.inp", "edb_2D_Spi05.inp",
"edb_ICOLD.inp", "edb_KarlSand.inp", "edb_Pon10.inp",
"rectangular_side_weir_Fr_0_491.inp", "wave_motion_for_WaveSAX.inp".

commit 98cee1d8218c2a09e70a8d396f7607a7dae48e67

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Mon Nov 27 11:22:37 2017 +0100

Minor modifications.

commit 94a99094f1d798e143eaa6f25acb7ad0fa58bbff

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Tue Sep 5 17:22:23 2017 +0200

Minor modifications.

SA_SPH. An omp critical section is inserted in the program unit
"ComputeBoundaryDataTab". The omp critical section "omp_FBCE3D" of the
program unit "FindBoundaryConvexEdges3D" is extended.

Main_algorithm. The omp critical sections "omp_Ncbf_Max" and
"omp_Ncbf_Max_2" are inserted in the program unit "Loop_Irre_3D".

commit ef8db5888fb2b76c7c62e6c55b1050a78b61cccc

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Tue Sep 5 09:57:23 2017 +0200

Minor modifications.

BC. Program unit "CancelOutgoneParticles_3D". The computation of a
line-plane intersection is corrected.

BC. Some omp critical sections are defined in the program units
"CancelOutgoneParticles_3D" and "CancelOutgoneParticles_2D".

Post_processing. The free surface detected by the monitoring lines is
increased of $dx/2$ to consider that the free surface is located at the
edge ($dx/2$ over the particle barycentre) of the detected particles.

Time_integration. The new program unit "time_step_duration" merges the old
program units "inidt2" and "rundt2".

BE_mass. The new program unit "Continuity_Equation" merges the old program
units "inter_EqCont_3D" and "inter_EqCont_2D".

BE_momentum. The new program units "velocity_smoothing",
"velocity_smoothing_SA_SPH_2D" and "velocity_smoothing_SA_SPH_3D" merge
the old program units "inter_SmoothVelo_2D" and "inter_SmoothVelo_3D".

Obsolete program units. The following obsolete program units are removed:

"I_O_ENG_module", "I_O_ITA_module", "I_O_language_module",
"SearchforParticleZone_3D", "ltrim".

commit 3e57ac797faf6b48d16735db4f39f711b25c7eac

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Tue Jun 13 15:39:10 2017 +0200

Minor modifications.

Documentation files. Sliding friction and normal reaction forces. On the
overall normal of the neighbouring frontiers.

commit 8b4a7d025d7b9fc4c3a11e25806fc9bc378e1589

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Tue Jun 13 15:17:45 2017 +0200

Minor modifications.

Body transport. Dry stage (no neighbouring fluid particles) of body-frontier interactions. Explicit formulation for the sliding friction force, depending on the friction angle between the computational body and the neighbouring frontier. The direction of the sliding friction force is the opposite to the velocity direction of the centre of mass of the computational body (projected on the local DEM plane). Present approximations: a unique friction angle applies to all the body-frontier interactions; the vector sum of the normal reaction force under sliding and the sliding friction force provide no contribution to the body torque (nevertheless the limiter for the sliding friction force depends on the velocity of the solid particles interacting with the frontiers). In case the input friction angle is negative, then sliding friction depends on the local slope/interface angle (this always applies for body-body interactions).

commit bf6ec5499859e874649a888007dfd2ac0255abcb

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Fri May 26 17:15:41 2017 +0200

Minor modifications.

Bug fixed. Post-processing. On the initialization of the arrays `q_max` and `Z_fluid_max` in a restarted simulation.

commit 5987d6bac30af5d8f29066e3c3f419bbf662f9b0

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Tue May 23 15:45:28 2017 +0200

Minor modifications.

SA-SPH. Input variable “`laminar_no-slip_check`” to enable/disable the local check on the laminar regime to eventually activate no-slip conditions. Formerly, “`laminar_no-slip_check`” was implicitly “`.true.`”.

commit cf90bc47da1f6ca87980d920b5621d13b5ee5b4c

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Tue May 23 11:35:55 2017 +0200

SA-SPH. No-slip conditions are activated in 3D (Di Monaco et al., 2011, EACFM): assessment in progress.

commit 177afd0d84fa7e07b8ad5ebe839e24b477ad1bf0

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Mon May 15 17:53:16 2017 +0200

Minor modifications.

Body dynamics. Documentation file. An appendix is added. Meaning of

the following approximated configuration of SPHERA:
(imping_body_grav=1, imping_body_grav_dry=0): clarifications on
gravity force, sliding friction force, body-boundary normal
reaction force under sliding, normal coefficient of restitution.

commit 211cd11398ae27e0ecb271932d8810602ee2daa4
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Tue Mar 28 16:33:37 2017 +0200

Minor modifications.

Body transport. Additional option to deactivate gravity during any dry
impingement.

commit 03fc89e6c807cef61341a84df82aee26231c446e
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Thu Mar 23 13:41:53 2017 +0100

Minor modifications.

Post-processing. Maximum body particle acceleration is available in the
application log.

Bug fixed. Body transport. Misprint on the definition of the variable
"imping_body_grav" on the template of the input file.

Bug fixed. Body transport. On the use of the variable
"time_max_no_body_gravity_force".

commit 338f8141e7ff06bdb656abd6de22dd6613fd31d1
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Wed Mar 22 14:31:43 2017 +0100

Minor modifications.

Body transport. FSI (Fluid-Structure Interaction). Both free-slip (new
formulation) and no-slip conditions (Adami et al., 2012) are available.
So far, no-slip conditions have been used.

commit 8f3cbaf810caa49b85213bcdfed8f440e36b7d5b
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Tue Mar 21 19:37:32 2017 +0100

Minor modifications.

Body transport. Limiter for maximum pressure values on the body surfaces.
This limiter only avoids extremely high and unphysical values.

commit d764eb10b7b19374e24f1cb8023a11ef02e1a0f1
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Wed Mar 15 11:26:07 2017 +0100

Minor modifications.

Body transport. Input variable to activate a limiter which avoids pressure
negative values on the body surfaces.

commit 9f12acfb42de0685fb7a22ea52cd7b7568ff9e5

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Tue Mar 14 17:25:28 2017 +0100

Minor modifications.

Body transport. Input variable “time_max_no_gravity_gravity_force”: gravity force is deactivated until this time. No deactivation in case of negative value.

Body transport. Input variable “time_max_no_body_frontier_impingements”: body-frontier impingements are deactivated until this time. No deactivation in case of negative value.

commit 3b76a72dd26d5d12c55810587081c2d3459764d3
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Tue Mar 14 15:32:14 2017 +0100

Minor modifications.

Bugs fixed. Pre-processing; Post-processing. On the restart procedure.

Bugs fixed. Post-processing. On the allocation of the arrays “q_max”, “h_step”, “qx_step”, “qy_step”, “h_max”.

commit f0272cf22412ffbca0f920f68e2cbe63ef9790af
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Fri Mar 10 18:21:45 2017 +0100

Minor modifications.

Body transport. I/O management of the body orientation to initialize a new simulation based on the data of a previous run (and with no need for a restart procedure). The IC rotation matrix (to set body particle IC) is based on Rodrigues’ formula (Euler’s theorem) and depends on the input data. The IC orientation of a generic body (and a generic body element) is provided in terms of rotation axis and rotation angle (instead of Euler angles; ref.: input file template). The rotation axis and the rotation angle are written in the output file “Body_dynamics.txt” and refer to the following times: initial time, current time. The vector “alfa” is not read/written anymore in I/O management. Implemented as an approximated procedure, the new modification only guarantees a correct reinitialization of the relative position of the first body particle. Nevertheless, as the body centre of mass is correctly reinitialized, the error in the body orientation (of a reinitialized simulation) is limited.

commit e3ca4bdfa16e632f58865a580481f1939a0d5298
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Tue Mar 7 13:36:51 2017 +0100

Minor modifications.

Bug fixed. Post-processing. Min./max. of the absolute value of the body angular velocity on the application log.

Body transport. Rotation matrix computation is formally more accurate. No influence on the test cases treated so far.

commit 7ffae125eb66f7da766d3193a1dee406934accd5

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Fri Mar 3 15:27:12 2017 +0100

Minor modifications.

Bug fixed. IC. Allocation of the auxiliary variable `z_aux`: dependency on the "fictitious vertex" .

commit 33ea4ab906d8c07e8c5bc5936b235c438ae879a3
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Fri Mar 3 11:38:02 2017 +0100

Minor modifications.

Body transport. The variable "alfa" (rotation angle of the body with respect to the reference system) is reported in the output file "Body_dynamics.txt" .

commit db85a1c659d75e327081babbf0a58c8c3a9c14f1
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Thu Mar 2 13:35:53 2017 +0100

Minor modifications.

IC. There is no constraint on the number of reservoirs which can be extruded from topography (only one extruded reservoir was permitted).

Pre-processing. The numerical parameters `nag_aux` (slight overestimation of the total number of fluid particles) does not depend anymore on the reservoir. Other minor modifications on the template of the input file.

Bug fixed. IC. The upper particle ID of a given zone/reservoir is computed. The parameter `nag_reservoir_CartTopog` is not used anymore.

commit 8e93fc0ab9884337e960f9c084d180cfb0448b6e
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Wed Feb 22 15:55:57 2017 +0100

Minor modifications.

"z-coordinate" is available in the ".vtu" output files, no matter about the configuration of the input files.

commit 221723dbaa0bad8ee33eaab6df505368e8bdfbeb
Merge: 83959e7 49c46ab
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Fri Feb 3 10:44:31 2017 +0100

Merge branch 'master' into SPHERA_v_8_0_AA

commit 49c46abdbfc251ef6f7080e7acfa99e2c02af52b
Merge: afc1a53 02e427f
Author: Andrea Amicarelli (RSE SpA) <Andrea.Amicarelli@rse-web.it>
Date: Fri Feb 3 10:46:36 2017 +0100

Merge pull request #2 from SauroManentiUNIPV/SPHERA_v_8_0_SM

Limiting viscosity

commit 83959e73823a71a1794471d64ca82c3d73792476
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Fri Feb 3 10:37:19 2017 +0100

Minor modifications.
Additional comments.

commit 02e427f68af69a7046cacb425377cec260996aba
Merge: 7161c7a afc1a53
Author: SauroManentiUNIPV <sauro.manenti@unipv.it>
Date: Wed Jan 25 10:23:45 2017 +0100

Manual merging of master into branch SPHERA_v_8_0_SM .

commit 7161c7a57e49a6c206f2042ce81506d12179992a
Author: SauroManentiUNIPV <sauro.manenti@unipv.it>
Date: Tue Jan 24 10:11:45 2017 +0100

Limiting viscosity.

commit afc1a5340b08fabdb664c1e33ab734931aada96e
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Mon Jan 23 16:34:48 2017 +0100

Minor modifications.

- Bug fixed. Body transport. On the reference system change in body-frontier interactions in case of triangular frontier faces.
- Bug fixed. On the allocation of the arrays of the maximum specific flow rate and the maximum water depth duiring a restart run.
- Bug fixed. DB-SPH. On the removal of the fictitious reservoirs at the beginning of the simulation.

commit 45fcb6b07116b84f18f2e8aebcbf7fde0af335e9
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Tue Jan 17 18:49:47 2017 +0100

Minor modifications.

- Post-processing. More inclusive format specifiers in the application log.

commit 291635f710350286d15d92685d9ca0d7809098f6
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Tue Jan 17 16:57:32 2017 +0100

Minor modifications.

- Corrections on the management of the body transport arrays during the restart procedures.

commit 2bd0c242874c61ff1e6828f534b3f80a16b7af11
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Tue Jan 17 15:31:54 2017 +0100

Minor modifications.

Bugs fixed. Misprints.

commit 329ab71887d99bdb6a8fd76bf4b1a4a7701e8484

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Tue Jan 17 15:06:26 2017 +0100

Minor modifications.

Bugs fixed. Checks on the allocation of some arrays for body transport.

commit 030b64e788b55403562d75dbb0e8d526b5121780

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Thu Dec 22 11:06:58 2016 +0100

Minor modifications.

Bug fixed (post-processing). Allocation of the array of the specific flow rate (q_{\max}) is function of the number of the topographic vertices (instead of the positioning grid vertices).

commit 4e7f0a6adef6c1490adffdd0c9e20975af3f8900

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Fri Nov 25 14:51:18 2016 +0100

Minor modifications.

New test case: ICOLD erosional dam break (on complex topography).

commit 4928313ac1ea3c46246944a8fbc8bdb9b39fad1

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Thu Nov 17 16:58:44 2016 +0100

Minor modifications.

Bug fixed. In the restart mode, the value of the variable “GCBFVecDim” provided by the input file is not used. Instead, the restart file provides the correct value.

Bug fixed. In the restart mode, the array “GCBFPointers” is correctly allocated.

commit 6d6dec91f1b7f3a626d93a4f8d2ced5114d83d1e

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Tue Nov 15 17:24:59 2016 +0100

Minor modifications.

Bug fixed. Variable “GCBFVector” is correctly allocated in the restart mode. Variable “NumBSides” is correctly written and read in the restart mode.

commit 894697dd17c970d3dd2d832197eb3f0757d92f17

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Wed Nov 9 12:00:49 2016 +0100

Minor modifications.

Bug fixed. New “if” constructs depending on the variable “diffusione” (subroutines “rundt2” and “inidt2”). The bug might have caused minor errors in estimating the time step duration (dt).

Bug fixed. Restored the mobile particle counting (to assess the variable “indarrayFlu”; initialization section of the subroutine “LoopIrre3D”).

Bug fixed. Detection of the elastic strain rate regime (subroutine “mixture_viscosity”).

commit b96ae71e82ebe0daaa28025ba7eb0d701feb3aa4

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Fri Oct 28 10:01:40 2016 +0200

Minor modifications.

Bed-load transport. Bug fixed in computing the mixture viscosity.

commit f328621c31a5ce4a98327f051a279122c8ea1fa2

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Thu Oct 27 17:13:23 2016 +0200

Minor modifications.

Bug fixed. Bed-load transport. The wrong treatment of local uniform velocity fields is deleted.

commit 84a39cbeea9b92e5101571ff53d689a4a1353b1b

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Thu Oct 27 12:25:04 2016 +0200

Minor modifications.

Bug fixed. Mixture particles in the elastic-plastic strain regime (mixture viscosity higher than the reference threshold): acceleration is zeroed in 3D by the subroutine “LoopIrre3D” (as in 2D; zeroing velocity is not sufficient). Notice that the involved particles are held fixed.

commit df636281b5fdb7a8d4d78a972e824f73f605e4c3

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Wed Oct 26 17:18:48 2016 +0200

Minor modifications.

Bug fixed. For those particles, which are held fixed, the dynamic viscosity is formally set equal to the maximum/threshold frictional viscosity (instead of fluid viscosity; subroutines “LoopIrre3D” and “LoopIrre2D”). Due to the presence of several “if” constructs, which are based on conditions depending on the particle viscosity, this modification may influence the computation.

Bug fixed. The subroutine “Shields” is now properly called when using RK time integration schemes (subroutine “LoopIrre3D”).

commit a49ded801956e8d1058be97bcbb0b4f488f802

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Mon Oct 24 17:31:34 2016 +0200

Minor modifications.

Bed-load transport. Particles modify their mobility status (fixed/"sol" or mobile/"flu") only in the subroutine "Shields" (not anymore in the subroutine "mixture_viscosity"), according to the fixed bed detection (in "CalcVarLength"), no matter about the cause of change (either the velocity threshold or the maximum frictional viscosity). So far, the momentum equation and the velocity smoothing procedures are skipped for mixture particles in the elastic-plastic strain regime. This is not due to the particle status (it is not "fixed" at this stage of the algorithm), but to the particle viscosity, which is at least equal to the frictional viscosity threshold ("if" condition on ...%mu = ...%mumx). This modification already began with the previous commit.

Bugs fixed.

commit 385cddf18b53d2f650017bfab06f833f474f8ea6

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Fri Oct 21 17:16:51 2016 +0200

Minor modifications. Auxiliary procedures are deleted. Notice that some SPHE pre-processing tools are available at <https://github.com/AndreaAmicarelli> (e.g., DEM2xyz -RSE SpA-, ply2SHERA_perimeter -RSE SpA-).

commit c6266690a785ae390bcd40f8ef79f7f5d3e0c652

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Fri Sep 23 18:19:56 2016 +0200

Minor modifications.

Bed-load transport. Restored the initial mobile particle counting in the subroutines Loop_Irre_3D and Loop_Irre_2D, in the presence of the boundary treatment SASPH.

Bed-load transport. The following redunant variables are erased:

Med()%modelloerosione, erosione, modelloerosione.

commit 7ad2b8c70b1f59afd966845884ddd2b01dd324ed

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Fri Sep 16 18:39:43 2016 +0200

Minor modifications.

commit 28aab0183d8277001ecd19103945c70cd70b13a7

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Tue Sep 13 08:07:28 2016 +0200

Minor modifications. The following test cases are published at SPHERIC 2016:

San_Fernando_Lower_van_Norman_dam_liquefaction.inp,

dike_breach_2D_expSchHag12JHR_ID13.inp,

erosional_dam_break_Karlsruher_sand.inp, spherical_Couette_flow.inp.

commit 293fa92960a8bbb4c56284883621a4d77a7d37ef

Merge: b0911f6 d7c4348

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Tue Sep 13 07:56:04 2016 +0200

Merge branch 'master' into SPHERA_v_8_0_AA

commit d7c43486d6e90f52ab00c5a7ae95d671a058cdf8
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Tue Sep 13 07:51:19 2016 +0200

input file for SPHERA v.8.0 (RSE SpA): Benchmark2_SASPH. Copyright 2005-2016
(RSE SpA; authored by Antonio Di Monaco, Giordano Agate, Andrea Amicarelli).

References: Di Monaco et al. (2011, EACFM), Amicarelli et al. 2013 (IJNME).

commit b0911f667eca5adc2adf1d33792bbf54259c09ad
Merge: 2f4307a 7db30ab
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Thu Sep 8 17:33:11 2016 +0200

Merge branch 'master' into SPHERA_v_8_0_AA

commit 7db30ab95c6a4b1508b4d37a1108edd1557495a7
Merge: 1a2e6e1 5e23337
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Thu Sep 8 17:22:36 2016 +0200

Merge branch 'master' of <https://github.com/AndreaAmicarelliRSE/SPHERA>

commit 2f4307a43bc2e9b94c49dc0945b189c715b12491
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Thu Sep 8 08:09:05 2016 +0200

Minor modifications.

commit 5e233370f4b5a885fbdc1dadde17dfb56600e48b
Merge: a7893bc f3041c0
Author: Andrea Amicarelli (RSE SpA) <Andrea.Amicarelli@rse-web.it>
Date: Mon Sep 5 11:02:15 2016 +0200

Merge pull request #1 from chqiao/patch-1

Minor modification. Format correction in the application log (for portability with gfortran compiler).

commit f3041c0219578af9dc7eb2f0e602395e9bef4450
Author: QIAO Cheng <qiaoch@yeah.net>
Date: Thu Aug 18 20:50:20 2016 +0800

Correct the write out format of headline

Correct the write out format problem of screen display which will make a error while compiling the code.

commit 1a2e6e1a61a836f31d84981e751ecb08e8b52c0e

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Mon Jul 4 14:19:57 2016 +0200

Time integration. Two time integration coefficients are requested in input to compute the time step duration: CFL and vsc_coeff (coefficient for the viscous stability criterion; default value: 0.05).

Restart procedure. In 3D, arrays GCBFVector & GCBFPointers are saved and read in/from the restart file, whereas subroutine

GridCellBoundaryFacesIntersections3D is not called during a restarted run.

Auxiliary procedure. ply2SPHERA_perimeter. An offset for z-coordinates can be imposed from the input file.

commit a7893bcc577c3747b77b259e02d47df679b3a679

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Mon Jun 13 13:17:55 2016 +0200

Minor modifications. Some new input files.

commit c6b5f503e88043c4c9a65e980732bf64b244c4c9

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Mon Jun 13 13:04:48 2016 +0200

Bed-load transport. Lagrangian saturation scheme. Modification to obtain null fluid pressure for those (possible) dry mixture particles, which lie below saturated mixture particles.

Minor modifications. New input files.

commit 032c13b87d9e1234ab46746d39703517d3ea54c1

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Wed Jun 8 11:48:08 2016 +0200

Bed-load transport. New conservative Lagrangian scheme for saturation conditions (hypothesis of stratified flows).

Mixture particles are either fully saturated or dry. The variable saturated_medium_flag is added as an input logical variable to indicate if a medium is fully saturated or dry. The mixture density is computed accordingly.

The variable alfa_WT (Water Table slope) is computed on the mixture particle, which locally defines position of the water table). The formulation for the fluid pressure depends on this angle, which approximately represents the underground flow direction.

ID of the highest saturated mixture particle in the background grid column. This local height is approximately representative of the top of the fully saturated zone (under the hypothesis of stratified flows). This parameter is stored in the array ind_interfaces(i_grid,j_grid,6).

Bed-load transport. The redundant input variable ID_granular is not requested anymore.

Post-processing. The subroutine write_Granular_flows_interfaces is split in write_Granular_flows_interfaces and interface_post_processing.

Bug fixed. Post-processing. Correction on format specifiers for the application log file.

Bug fixed. Post-Processing. The subroutine cat_post_proc is now called by the subroutine result_converter. In other words, the .txt files are concatenated each time new .vtu output files are written.

commit acc0448528c779745f4b6b1cf603680f2d16ed88

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Thu May 26 11:25:23 2016 +0200

Minor modifications.

commit e5363e585c2c4d10bf60ff0ad02fe90fb802ab44

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Thu May 26 08:35:40 2016 +0200

Post-processing. File concatenation of output .txt files is carried out incrementally at each post-processing stage. Thus, the .txt output files are concatenated even in case of a simulation stop or kill. New subroutine: cat_post_proc.

Post-processing. PV and log files allow using a number of particles higher than 999'999.

commit 210d87f88e2d8c947ac046f06388bb12bdb0eae3

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Tue May 24 10:05:23 2016 +0200

Bug fixed. IC. The check on the particle array dimension (subroutine SetParticles) is fixed.

commit 7beedef63de0f201c4620d65ee08d855fa206974

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Tue May 24 08:20:31 2016 +0200

Bug fixed. Geometry. Variable dis2 is correctly initialized in the subroutine point_inout_convex_non_degenerate_polygon. There were issues when the variable dis was null.

commit 02123239afeb56f4c77c12a7b05f308542bf2d9b

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Fri May 20 18:44:50 2016 +0200

Bug fixed. Auxiliary procedure ply2SPHERA_perimeter. Vertex IDs in face definition are corrected with the proper offset.

Bug fixed. Bed-load transport. Granular_flows_options%saturation_conditions is treated element per element, with no array notation (typo in CalcVarLength).

commit 4f88ff84a3a1039ac5929acd3aa5961283210f23

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Tue May 17 18:11:37 2016 +0200

Bug fixed. Geometry. Invalidated vertex IDs of SA-SPH faces can have both null and negative values (subroutine LocalNormalCoordinates).

commit 31dd699717332659fb3624c269613d34a620d819

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Wed May 11 16:11:00 2016 +0200

Bug fixed. Invalidated vertex IDs of SA-SPH faces can have both null and negative values (subroutines ModifyFaces, IsParticleInternal3D).

commit e0781e73493bdf9883e8b0b6b72ee0d05a984cf3

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Wed May 11 11:36:10 2016 +0200

Bug fixed. Restart. Arrays Z_fluid_max, q_max, Granular_flows_options%minimum_saturation_flag and Granular_flows_options%maximum_saturation_flag are allocated after reading their dimensions from the restart file (ReadRestartFile).

New input files: Benchmark1_dam_break_DBSPH, Benchmark2_DBSPH, dam_break_body_UniBas, erosional_dam_break_2D_FraCap02_Taipei, erosional_dam_break_bed_2D_FraCap02, erosional_dam_break_bed_2D_Spi05, erosional_dam_break_Pon10_asym, flushing_2D_small_granular_flows, flushing_2D_small_granular_flows_erosion_criterion_2D_complete, flushing_3D_small_slope_B3_granular_flows_erosion_criterion_2D_complete, flushing_3D_small_slope_B3_granular_flows_erosion_criterion_3D_complete, ICOLD_earth-fill_dam_breach_long, ICOLD_earth-fill_dam_breach_short, ICOLD_earth-fill_dam_break, sloshing_tank_TbyTn_0_78, sloshing_tank_TbyTn_1_07, submerged_landslide.

commit 1c0f2e717dc7839f5c483bb9272a426bcf483de0

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Thu May 5 20:46:05 2016 +0200

Bug fixed. IC. Variable dx_CartTopog is double precision, not an integer (issue with gfortran).

Bug fixed. IC. Variable h_reservoir (subroutine GeneratePart) was wrongly declared and initialized.

commit 2691ab8ae7bf45fa114769a9de065d298258fa72

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Wed May 4 17:29:38 2016 +0200

Minor modifications.

commit 73fdcebece3447174d1f7c93a02fcac8c55e0ad9

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Wed May 4 13:18:45 2016 +0200

Bug fixed. SA-SPH. Run-time reduction of the number of nodes (local variable

nnodes in the subroutine DefineLocalSystemVersors) in 3D.

commit 7649c717d6d8ab257c14a306d31af88dd1c99685

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Tue May 3 16:25:49 2016 +0200

Bed-load transport. Additional saturation schemes: uniform dry soil (saturation_scheme=0), uniform fully saturated soil (saturation_scheme=1). The saturation scheme, which depends on time_minimum_saturation and time_maximum_saturation, is represented by saturation_scheme=2.

Bed-load transport. The mixture top interface normal is computed depending on the mixture neighbours (instead of the fluid neighbours, which may be absent) and a default vertical vector is added (in the absence of mixture neighbours).

commit 04548f7c25312da841f83ad77b403b03969e16a3

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Mon May 2 15:00:35 2016 +0200

Minor modifications.

commit 6afe37c35135b449999a8817ddfd1cf64e686bf9

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Mon May 2 14:44:36 2016 +0200

Minor modifications.

commit e0970cddff774ceb89d6eff8d140eaf04c0f9c11

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Mon May 2 14:41:29 2016 +0200

Upgraded documentation of SPHERA v.8.0

commit 5b638350b97e3445654a441a37a2189e950cbe56

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Thu Apr 28 11:29:06 2016 +0200

Bed-load transport. New saturation conditions. The fluid pressure formula depends on the zone: phreatic zone, dry soil or infiltration zone. The infiltration front linearly grows between its height at the time with minimum saturation and those at the time with maximum saturation. The local soil bottom is detected as an interface. So far, the time at minimum saturation has to be smaller than (or equal to) the time at maximum saturation. When $t \leq t_{\min_sat}$, there is always phreatic zone below the free surface and dry soil elsewhere. When $t \geq t_{\max_sat}$, the saturation zones are freezed at t_{\max_sat} .

commit 217903d6caaae919c840640460ed3b87fb664f16

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Wed Apr 27 11:29:40 2016 +0200

Bug fixed. Bed-load transport. Granular_flows_options%saturation_flag is treated by the restart procedures.

commit bcc458b5bb7261926fa3fc3df80ed85e69c9edda
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Wed Apr 20 20:01:26 2016 +0200

Minor Modifications.

commit d4486112dc3a303a5e7baf648db6fa529a0aa1c2
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Tue Apr 19 17:04:27 2016 +0200

Bed-load transport. Saturation state is computed up to the saturation freezing time (input parameter).

Formal modifications. Variable it_corrente is named on_going_time_step. Variable tempo is named simulation_time.

commit 762d6356742461f6fd7f63a3bea8b3569bb9f8b8
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Mon Apr 18 17:39:04 2016 +0200

Bug fixed. Bed-load transport. Corrected the threshold constraint for the angle alfa_TBT.

Bug fixed. Bed-load transport. Nearest neighbour searching is mandatory even in the absence of any erosion criterion (for fluid phase pressure).

commit 7fef54266c4fe19ff36f231efd689f5487f64fc7
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Fri Apr 15 19:32:41 2016 +0200

Bed-load transport. Improvement of the formula for fluid phase pressure (simplifying assumption: 1D filtration with piezometric lines in the mixture parallel to the local 3D slope of the mixture top).

Bed-load transport. Computation of the mixture top interface normal.

Bug fixed. Bed-load transport. The interfaces between mobile and fixed particles are now computed only in the presence of an erosion criterion.

commit 505f3368f95bee0f4c48659960c54ffb25af4d61
Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>
Date: Fri Apr 15 10:34:17 2016 +0200

Bed-load transport. The mean effective stress is now estimated as a difference between the total stress and the pressure of the liquid phase. In general, xyz axis are not the principal axis and the lateral pressure coefficient at rest (K_0) only represents granular material at rest). So far, the model assumes hydrostatic conditions for the liquid pressure in the mixture. Mean effective stress is zeroed if the computation provides a negative value.

Bed-load transport. The fluid particle ID on the fluid side of the bed-load transport layer top interface is now computed even without any erosion criterion. This ID is mandatory to estimate the fluid pressure within the

mixture.

commit a1c505d19ed8c738bf5b11e4cf24cada9f70ba2e

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Thu Apr 14 12:29:53 2016 +0200

Bed-load transport. The formula for the mean effective stress is corrected according to the plane strain hypothesis of Mohr-Coulomb-Terzaghi criterion.

commit 2faf3a21338281ad9d7bfef65a4ecf3a9aaffabe

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Tue Apr 12 11:35:29 2016 +0200

Bug fixed. Bed-load transport. In the presence of mixture particles interested by the frictional viscosity threshold, $pg(\dots)\%var=0$ was missing when smoothing velocities (both in 2D and 3D). For safety reasons, now the variable `vel` replaces the variable `var` in time integration for trajectories (Leapfrog scheme). However, at that point, the variable `var` should be equal to the variable `vel` (in any case).

commit 4c9b23405a7d940dd9edeaba7dd46e9881f4de50

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Fri Apr 8 11:46:26 2016 +0200

Bed-load transport. The mixture particles, which are affected by the frictional viscosity threshold, are blocked.

commit 85f1ed9a63281a3321ac1a4d97c429b9cea9d258

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Tue Apr 5 17:23:11 2016 +0200

Bed-load transport. Contribution of the fluid phase to the frictional viscosity in the bed-load transport layer, just to avoid null values for the mixture particles at the layer top. Elsewhere, this contribution is negligible.

Bug fixed. Erosion criterion. Corrected the "if construct" in case the critical shear stress is close to zero.

commit 0ceee40754870bca2da34b939d023942092f22e3

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Mon Apr 4 17:30:44 2016 +0200

Bed-load transport. Frictional viscosity is the mixture viscosity in the bed-load transport layer.

Bed-load transport. The mean effective stress replaces the vertical effective stress, by means of the coefficient of lateral earth pressure at rest.

Bed-load transport. Formulations erased: Chauchat & Médale blt viscosity, Einstein dilute viscosity, Chezy-like viscosity. Parameter erased: `Granular_flows_options%viscosity_blt_formula`, `Granular_flows_options%Bn`, `Granular_flows_options %Chezy_friction_coeff`. Parameters erased in the input file: `Granular_flows_options%viscosity_blt_formula`, `Granular_flows_options %Chezy_friction_coeff`.

Bed-load transport. In case of null strain rate tensor (uniform flow or still material), then $dt_{\text{au}}/dx=0$. This condition is fictitiously represented with a null friction viscosity.

Bed-load transport. A viscosity threshold (μ_{max}) is used as a limiter.

commit 170c499e99d49264f3dce944b281c684a8862fe2

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Tue Mar 22 17:40:32 2016 +0100

Bug fixed. Restart: management of the variable Domain%start in ReadRestartFile.

Bug fixed. Restart: corrected file format o read the restart steps before the requested one.

Bug fixed. Restart: only the sections “domain”, “boundaries“, “vertices” and “faces” (main input file) are not read during a restart execution.

Bug fixed. Restart: correction on the first call of sub_Q_sections in case of restart.

commit e5c50c6401eceaafb241db2ff1cc3a5a92ddb5e7

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Mon Mar 21 18:38:39 2016 +0100

Minor modifications.

commit cd7627020ff22d5180d11d8b8803af0038c91c4c

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Thu Mar 17 13:33:30 2016 +0100

Minor modifications.

Bug fixed. Bed-load transport. Deactivation of the approach for liquefaction if the liquefaction time equals zero (approach deactivation).

commit aad4a6e4c6753c5aca70598d8b6acbecaed44737

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Wed Mar 16 17:54:31 2016 +0100

Minor modifications.

commit a10f63b8a3aa8e7f2e8e1b6b740a78a14db76ebc

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Tue Mar 15 17:31:17 2016 +0100

Bed-load transport. Liquefaction only applies, in the presence of the free surface along the vertical of the computational particle.

Bed-load transport. Effective stress is equal to the total pressure, in the absence of the free surface along the vertical of the computational particle.

Bug fixed. BC, DB-SPH. Argument of wavy_inlet is the inlet section ID, not the boundary ID.

Bug fixed. Bed-load transport. Calling Shields in LoopIrre_2D for DB-SPH (with bed-load transport, even without erosion criterion). The corresponding 3D correction is still missing.

Bug fixed. DB-SPH. Assigning surface_mesh_file_ID to inlet and outlet DB-SPH

elements.

Bug fixed. DB-SPH. Zeroed variables Ncbs, IntNcbs and Ncbf when using DB-SPH.

Bug fixed. DB-SPH. SPH approximation of density with NMedium>1.

Bug fixed. DB-SPH. surface_mesh_file_ID was not assigned in 2D (Import_ply_surface_meshes).

Bug fixed. Erosion criterion, DB-SPH. Mobile particles are counted after having removed DB-SPH fictitious reservoir (once they were set equal to the the number of particles in the domain).

Bug fixed (timing).

Tests (AA!!!) on multi-fluid DB-SPH.

commit 29a207b9fab8d8c5eb40cf584b39166f27dd2f26

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Thu Feb 18 18:59:52 2016 +0100

SPH-bl-3Dbd.

Liquefaction (preliminary implementation). Linear pore pressure functions (with respect to the cyclic number ratio).

commit f58226d86df2e433992cc6e8e27d681aff085b67

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Wed Feb 17 17:39:57 2016 +0100

SPH-bl-3Dbd: validations.

Minor modifications

DB-SPH. Graphical issue: avoided divisions by the discrete Shepard coefficient for non-wet wall elements.

commit 53b464b89df692a5231ed578641928fd2c0c70f7

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Tue Feb 16 17:42:45 2016 +0100

SPH-bl-3Dbd: validations.

DB-SPH. New input logical variable: negative_wall_p_allowed (pressure of wall elements can be negative).

DB-SPH. New input logical variable FS_allowed (free surface detection can be avoided).

commit 29bbc8b7930280cf14d27d02a8a3639b6d108e6c

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Fri Feb 12 19:26:04 2016 +0100

SPH-bl-3Dbd: validations. Minor modifications.

commit ed52bd4c8c23bf34a3cdb5ddd2ac48a217dff921

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Thu Feb 11 17:42:00 2016 +0100

SPH-bl-3Dbd: validations.

Momentum equation. Background pressure (Domain%prif) is now preliminarily implemented (it was only treated in pressure smoothing).

SPH-bl_t. The recently increase of $dx/2$ in the interface height (to compute effective pressure) is removed.

Bug fixed (pressure smoothing for 3D DB-SPH). This correction would improve the results of Amicarelli et al. (IJNME, 2013) on two 3D dam breaks.

commit fb5f6747b565785945171de5f5ba1ff7b60e9039

Author: AndreaAmicarelliRSE <Andrea.Amicarelli@rse-web.it>

Date: Thu Feb 4 18:32:11 2016 +0100

SPH-bl_t-3Dbd: validations.

Neighbouring search. Warning alert/ Error, in case the number of wall element neighbours is equal to / greater than the maximum number allowed.

DB-SPH. The upper limiter of the integral Shepard coefficient can be removed by means of an input variable.

Bug fixed (DB-SPH-NS-bl_t). Corrected the position of the line on: Shepard correction for the velocity gradient (times shear viscosity) of the semi-particles in the Viscous Sub-Layer of the Surface Neutral Boundary Layer.

Previously, it was erroneously executed for each fluid particle neighbour.

commit b1c20b4708ff71009fa3f8dca4f06f2a7765c860

Author: AndreaAmicarelliRSE <andrea.amicarelli@rse-web.it>

Date: Fri Jan 29 17:53:00 2016 +0100

SPH-bl_t-3Dbd: validations.

SPH-bl_t-3Dbd. New variable DBSPH%slip_ID.

SPH-bl_t-3Dbd. Ferrand et al. (2013, IJNMF) formulation: BC contributions (both terms) to viscosity shear stress term are null if the fluid particles are still. The formulation is adapted to moving boundaries.

SPH-bl_t. Added $dx/2$ to the top of the bed-load transport layer to improve the estimation of the effective stress.

Bugs fixed (DBSPH_kinematics). A parallel do end was missing. Variables vel_aux and omega_aux are now shared variables in every omp cycle.

Bug fixed (Loop_Irre_3d and Loop_Irre_2d). Wrong if construct (accelerations were always null with DB-SPH). This bug was introduced after SPHERA v.8.0 and has not affected any study/release.

Bug fixed (Time_integration). Corrected estimation of dt in the presence of only inlet sections and mixture reservoirs.

commit 1be37160f3fe0bd4096da30dd4ba2293b482f76b

Author: AndreaAmicarelliRSE <andrea.amicarelli@rse-web.it>

Date: Wed Jan 27 19:03:15 2016 +0100

SPH-bl_t-3Dbd: validations.

Minor modifications (the bugs do not affect the results of previous studies):

Semi-particle (DB-SPH) viscosity is now normalized by means of the discrete Shepard correction.

Bug fixed on time integration. Neglecting mixture particles in CFL computation considered every non-fixed particle, fluid particles included.

Bug fixed on DB-SPH. vel_aux and omega_aux were used even in case of no imposed kinematics (and were not initialized). Now an “if construct” only selects wall elements related to a boundary with imposed kinematics (no need for

initializing vel_aux and omega_aux).

Bug fixed on DB-SPH. vel_aux and omega_aux were private (instead of shared) in a parallel do.

Bug fixed on time integration (initialization of done_flag in 3D).

commit d6b2c2b4a506711b0f17c232ef774c485a126c33

Merge: 299175c 55981ad

Author: AndreaAmicarelliRSE <andrea.amicarelli@rse-web.it>

Date: Mon Jan 25 10:29:33 2016 +0100

SPH-bl-3Dbd. Merging from master.

commit 299175c55faa00e40b995071e0f9af9c2a78773c

Author: AndreaAmicarelliRSE <andrea.amicarelli@rse-web.it>

Date: Thu Jan 21 17:23:37 2016 +0100

SPH-bl-3Dbd: validation in progress. Test removed.

commit 9d138d56bce8b1d225f8aa226ade45e48f76f13e

Author: AndreaAmicarelliRSE <andrea.amicarelli@rse-web.it>

Date: Thu Jan 21 17:19:33 2016 +0100

DB-SPH-NS-bl (or equivalently SPH-bl-3Dbd: SPH - bed-load transport - 3D bottom drag). Validation in progress. Minor errors fixed.

commit f4d8f7fdf8c9853eafc5b78b217c5d255dfc8d4b

Author: AndreaAmicarelliRSE <andrea.amicarelli@rse-web.it>

Date: Wed Jan 20 18:44:52 2016 +0100

Merging issues completed. Fixed the merging conflict errors accidentally introduced in commit of 29Oct15, 10h26.

commit 403c02b80da9cc742c97910c91719ee28d7c96eb

Author: AndreaAmicarelliRSE <andrea.amicarelli@rse-web.it>

Date: Wed Jan 20 14:52:19 2016 +0100

Merging issues. In progress.

commit 141f00c910da228cf71c3232ef28ab783db8d265

Author: AndreaAmicarelliRSE <andrea.amicarelli@rse-web.it>

Date: Tue Jan 19 18:35:52 2016 +0100

Merging issues. In progress.

commit 55981ad8dacb54a5a2c204edbcca78f9bcb5715

Author: AndreaAmicarelliRSE <andrea.amicarelli@rse-web.it>

Date: Thu Nov 19 18:22:05 2015 +0100

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

13. USER GUIDE

SPHERA installation is straightforward (Sec.13.1), 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.13.2). This template defines and comments all the input parameters. Finally, SPHERA v.9.0.0 tutorials are discussed in Sec.13.3.

13.1. Installation

SPHERA source and executable files are distributed on a dedicated Git repository on GitHub ([184]). 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").

13.2. Commented template of the main input file of SPHERA v.9.0.0

Figure 13.1 reports the commented template of the main input file of SPHERA v.9.0.0.

The comments define all the input parameters and describe the meaning of their possible values. Further, suggested or default values are reported.

```
version.subversion.subsubversion  !
                                ! SPHERA main input file: template and
                                ! comments
!-----
! SPHERA v.9.0.0 (Smoothed Particle Hydrodynamics research software; mesh-less
! Computational Fluid Dynamics code).
! Copyright 2005-2018 (RSE SpA -formerly ERSE SpA, formerly CESI RICERCA,
! formerly CESI-Ricerca di Sistema)
!
! SPHERA authors and email contact are provided in SPHERA documentation.
!
! This file is part of SPHERA.
! SPHERA is free software: you can redistribute it and/or modify
! it under the terms of the GNU General Public License as published by
! the Free Software Foundation, either version 3 of the License, or
! (at your option) any later version.
! SPHERA is distributed in the hope that it will be useful,
! but WITHOUT ANY WARRANTY; without even the implied warranty of
! MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
! GNU General Public License for more details.
! You should have received a copy of the GNU General Public License
! along with SPHERA. If not, see <http://www.gnu.org/licenses/>.
!-----
##### 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),bsph(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,
! and does not belong to any boundary (only useful for Paraview)
1 Vertex_x Vertex_y Vertex_z ! (first vertex data)
! ... ! (other vertices; no gaps on vertex
! IDs)
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 "fictitious 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 (no gaps on line
! IDs)
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 "fictitious reservoirs".
! In case of SA-SPH boundary treatment scheme, the code requires the
! the actual 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 1/2/3 vertex IDs are 0 in case of pentagonal/quadrilateral/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 clockwise to keep coherence with SA-SPH walls).
! Faces can only be triangles or rectangles (generic quadrilaterals are not
! permitted), but only "perimeter" faces can also be generic quadrilaterals (not
! only rectangles), pentagons and hexagons.

```
1 face_1_vertex_1 ... face_1_vertex_6 face_1_Boundary_ID
                        ! first face data ("-1" is mandatory
                        ! for meaningless vertex IDs (e.g.,
                        ! 5th and 6th vertices of a
                        ! quadrilateral)
...                      ! other records; no gaps on face IDs)
face_last_ID face_last_vertex_1 ... face_last_vertex_6 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(only
 ! removal of the particles crossing the boundary),
 ! tapis (not recommended)

! If (Boundary_type=="fixed"): start
Shear_stress_coefficient laminar_flag_check ! Shear_stress_coefficient=1.0
 ! (no-slip, under assessment in 3D
 ! with SA-SPH),0.(free-slip,
 ! default choice in 3D with SA-SPH);
 ! laminar_no-slip_check=.true.(check
 ! on laminar regime to activate
 ! shear stress terms at boundaries
 ! in case of no-slip conditions;
 ! laminar regime check has no
 ! effect in the inner domain),
 ! .false.(no check: always apply
 ! no-slip conditions in case
 ! Shear_stress_coefficient=1.0)

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 DBSPH_fictitious_reservoir_flag
                           ! 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);
                           ! DBSPH_fictitious_reservoir_flag = .true.(DB-SPH
                           ! fictitious fluid particles to complete the kernel
                           ! support at the free-surface, in pre-processing),
                           ! .false.(no fictitious fluid particles)
! If (IC_reservoir_type==2): start
dx_CartTopog H_res        ! dx_CartTopog(spatial resolution of the Cartesian
                           ! topography; dx_CartTopog>=dx); 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                  ! n_circum(number of vertices circumscribing
                           ! the horizontal projection of the reservoir)=3,4
circum_1_x circum_1_y      ! First point of the 2D figure circumscribing the
                           ! horizontal projection of the reservoir. Admitted
                           ! figures: triangles and convex non-degenerate
                           ! quadrilateral (otherwise the subroutine call for
                           ! point_inout_convex_non_degenerate_polygon must be
                           ! replaced with other calls to subroutines such as
                           ! point_inout_quadrilateral).
...                          ! 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. Admitted
          ! figures: triangles and convex non-degenerate
          ! quadrilateral (otherwise the subroutine call for
          ! point_inout_convex_non_degenerate_polygon must be
          ! replaced with other calls to subroutines such as
          ! point_inout_quadrilateral). From a plan view, the
          ! vertices are provided anticlockwise.
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
fluid_ID
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 slip_ID Gamma_limiter_flag
! 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)
! slip_ID (ID for slip conditions) = 0 (free-slip), 1 (no-slip),
! 2 (run-time choice depending on the inner shear viscosity terms in SPH-NS
! balance equations)
! Gamma_limiter_flag: logical variable to activate or deactivate Gamma upper
! limiter (.true.)
negative_wall_p_allowed FS_allowed ! negative_wall_p_allowed: pressure of wall
          ! elements can be negative (logical);
          ! FS_allowed: free surface detection can be
          ! avoided (logical).
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
surface_mesh_files flag_in-built_monitors ! surface_mesh_files: number of files
          ! of the DBSPH surface meshes;

```



```

! flag_in-built_monitors(logical):
! flag for in-built motion of control
! lines and DB-SPH frontiers
! do i=1,surface_mesh_file
imposed_kinematics_i_records rotation_centre_i_x rotation_centre_i_y rotation_centre_i_z
! imposed_kinematics_i_records(number of
! records, which describe a possible imposed
! kinematics for the i-th DBSPH surface
! mesh file);
! rotation_centre_1 (centre of rotation for
! DB-SPH frontiers for the i-th DBSPH
! surface mesh file)
! if (imposed_kinematics_i_records>0): start
i_time_1 i_velocity_x_1 i_velocity_y_1 i_velocity_z_1 i_omega_x_1 i_omega_y_1 i_omega_z_1
! ... ! other possible records
i_time_last i_velocity_x_last i_velocity_y_last i_velocity_z_last i_omega_x_last i_omega_y_last
i_omega_z_last
! (records for the imposed
! kinematics to frontiers for the i-th
! DBSPH surface mesh file); time of the last
! record should be slightly higher than time
! at the end of the simulation.
! if (imposed_kinematics_i_records>0): end
! enddo
n_inlet n_outlet ply_n_face_vert ! n_inlet(number of inlet sections)
! n_outlet(number of outlet sections)
! ply_n_face_vert(maximum number of vertices
! of the DB-SPH faces as represented in the
! ".ply" input files(3/4/5/6 in 3D, 4 in 2D)
! if (n_inlet>0): 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>0): end
! if (n_outlet>0): 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>0): end
##### END DBSPH #####

##### BED LOAD TRANSPORT #####
! Input parameters for bed-load transport (blt) scheme

```

```

erosion_criterion_ID ID_main_fluid
    ! 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).
saturation_scheme time_minimum_saturation time_maximum_saturation
    ! saturation_scheme=0(dry soil),1(fully
    ! saturated soil),2(saturation zones
    ! depending on time_minimum_saturation and
    ! time_maximum_saturation),3(Lagrangian
    ! scheme for saturation conditions; mixture
    ! particles are either fully saturated or
    ! dry).
    ! time_minimum_saturation: time related to
    ! a relative minimum saturation of the
    ! granular material.
    ! time_maximum_saturation: time related to
    ! a relative maximum saturation of the
    ! granular material. So far,
    ! time_minimum_saturation has to be smaller
    ! than (or equal to)
    ! time_maximum_saturation. When  $t \leq t_{\text{min\_sat}}$ 
    ! , there is always phreatic zone below the
    ! free surface and dry soil elsewhere. When
    !  $t \geq t_{\text{max\_sat}}$ , the saturation zones are
    ! freezed at  $t_{\text{max\_sat}}$ .
! 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)
deposition_at_frontiers Gamma_slope_flag
    ! 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)
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

```

```

t_q0 t_liq          ! t_q0: quake start time; t_liq:
                    ! liquefaction time
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(only if
                    ! erosion_criterion_ID>0)
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    ! >0.
roughness_coefficient ! null recommended value (i.e. inactive
                    ! parameter)
! If (fluid_type==liquid): end
! If (fluid_type==granular): start
phi saturated_medium_flag ! phi(internal friction angle in degrees,
                    ! even if the code works in radians);
                    ! saturated_medium_flag=.true.(fully
                    ! saturated medium),.false.(dry medium).
cohesion viscosity_max tuned_viscosity limiting_viscosity
                    ! cohesion (draft parameter); viscosity_max
                    ! (threshold for the dynamic mixture
                    ! viscosity to held particles fixed in
                    ! the elastic-plastic strain regime);
                    ! tuned_viscosity (tuned dynamic viscosity -
                    ! only for Manenti et al., 2012)
                    ! limiting dynamic viscosity
! If (fluid_type==granular): end
! if ((fluid_type==granular).and.(erosion_criterion_ID==1)): start
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
roughness_coefficient d_50
                    ! roughness_coefficient; d_50

```

```

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
! Bodies with imposed kinematics are listed after all the bodies with computed
! kinematics
n_bodies                dx/dx_body                friction_angle                time_max_no_gravity_force
time_max_no_body_frontier_impingements                body_minimum_pressure_limiter
body_maximum_pressure_limiter FSI_free_slip_conditions
                        ! n_bodies(number of transported solid
                        ! bodies); dx/dx_body(ratio between fluid
                        ! particle size and body particle size);
                        ! friction_angle(in radians; non-negative
                        ! value: friction angle for body-frontier
                        ! interactions; negative value: sliding
                        ! friction depends on the local slope angle
                        ! instead of the friction angle);
                        ! time_max_no_gravity_force(gravity
                        ! force is deactivated until this time, no
                        ! deactivation in case of negative value);
                        ! time_max_no_body_frontier_impingements
                        ! (body-frontier impingements are
                        ! deactivated until this time, no
                        ! deactivation in case of negative value);
                        ! body_minimum_pressure_limiter=
                        ! .true.(no negative pressure on the body
                        ! surface),.false.(no limiter);
                        ! body_maximum_pressure_limiter=
                        ! .true.(maximum pressure on the body
                        ! surface is limited by physical
                        ! constraints),.false.(no limiter);
                        ! FSI_free_slip_conditions=.true.(free-slip
                        ! conditions for FSI),.false.(no-slip
                        ! conditions for FSI)

! 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
n_R_IO_x n_R_IO_y n_R_IO_z tetra_R_IO

```

```

! n_R_IO(rotation axis -unit vector-);
! teta_R_IO(rotation angle). Rotation of
! the body with respect to the reference
! system at t=0. This rotation is
! cumulative with respect to the element
! initial rotation. This rotation is
! relevant only for IC and I/O purposes.
pos_rotC_x pos_rotC_y pos_rotC_z ! pos_rotC(centre of rotation for the
! vector alfa 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 describe the imposed kinematics)

! do i=1,n_records
time_i velocity_x_i velocity_y_i velocity_z_i omega_x_i omega_y_i omega_z_i
! i-th record for the imposed
! kinematics of the body (the time of the
! last record should be slightly higher
! than time at the end of the simulation)

! enddo
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)
n_R_IO_elem_x n_R_IO_elem_y n_R_IO_elem_z teta_R_IO_elem
! n_R_IO_elem(rotation axis -unit vector-);
! teta_R_IO_elem(rotation angle). Rotation
! of the body element with respect to the
! reference system at t=0. This rotation is
! cumulative with respect to the body
! IC rotation. This rotation is relevant
! only for IC and I/O purposes.
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)) for 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; in
                                ! case of a restarted simulation, these
                                ! values are reported in the log file, but
                                ! they are actually overwritten by the
                                ! values read from the input restart file);
                                ! this line has no influence in case of
                                ! restart
CFL vsc_coeff Leapfrog_flag scheme_order factor dt_alfa_Mon
                                ! CFL;
                                ! vsc_coeff(viscous stability condition
                                ! coefficient, default value: 0.05);
                                ! 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,
                                ! COEFNMAXPARTI considers that the number of
                                ! particles inside the domain can be higher
                                ! than the first assessment of the number of
                                ! SPH particles (initial particles or
                                ! reservoir particles -in case reservoir is
                                ! derived from topography-): COEFNMAXPARTI
                                ! takes into account contributions from
                                ! inlet sections and, in case of reservoir
                                ! derived from topography, also the
                                ! particles of the mixture dam/reservoir
                                ! (beyond the main water reservoir);
                                ! COEFNMAXPARTJ:maxb(max number of
                                ! neighbours)=COEFNMAXPARTJ*(4h/dx)^D (the
                                ! same number applies both for fluid
                                ! neighbours and DB-SPH wall element
                                ! neighbours;

```

```

! body_part_reorder(DB-SPH)=0(fixed
! frontiers),1(mobile frontiers)
nag_aux MAXCLOSEBOUNDFACES MAXNUMCONVEXEDGES GCBFVecDim
! nag_aux(rough overestimation of the
! number of fluid particles in the
! domain, it is only related to the very
! first allocation of the particle array,
! it is influential only in case of
! reservoir extruded from topography);
! MAXCLOSEBOUNDFACES(max number of
! neighbouring boundary face per fluid
! particle); MAXNUMCONVEXEDGES(max number
! of edges); GCBFVecDim (rough
! overestimation of the number of Grid Cell
! - Boundary Face intersections (SA-SPH;
! in case of restart, this value is only
! read, not used; in case its value is too
! little, crashes occur without code
! warnings/errors; GCBFVecDim only occupies
! 4B times the number of elements, thus, it
! is sufficient to overestimate this number
! as the number of faces times the number
! of the positioning grid cells, further,
! reducing GCBFVecDim does not imply any
! appreciable variation in the computational
! time).
density_thresholds_flag ! density_thresholds_flag=0(default, no
! density limiters),1(density limiters for
! debug)- Density limiters do not apply to
! the mixture particles which are held
! fixed (elasto-plastic strain rate
! regime)
##### 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 #####
restart_mode restart_time_value ! restart_mode("time" or "step");
! restart_time_value(positive time value in
! seconds or step number, according to
! restart_mode; this value has to be lower
! than the time value of the last restart

```

```

                                ! output; default value: 2.*dt_out_PV)
restart_file                    ! restart_file(restart file name -with
                                ! file extension-)
##### 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 for the water
                                ! front: output file ".fro")
restart time dt_restart        ! dt_restart(restart time frequency)
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_reservoir_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_reservoir_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 ! The edge order is not arbitrary to detect the
                                ! free surface. Edge 1 is the farthest from the
                                ! free surface.
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. These sections must
! be convex non-degenerate quadrilaterals. Otherwise, the subroutine call for
! point_inout_convex_non_degenerate_polygon must be replaced with other
! calls (to subroutines such as point_inout_quadrilateral).
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 ! to be omitted in case of triangles
                                ! vertices of the first section (
                                ! anti-clockwise viewing the section face
                                ! from the outside of the fluid domain)
...                            ! other possible section records
##### end section flow rate #####

##### substations #####
! Input variables for monitoring the electrical substations (both high-voltage
! transmission substations, medium-voltage distribution substations and
! low-voltage distribution substations). From a plan view, each substation is
! described by a polygon (either triangle, quadrilateral, pentagon or hexagon).
n_sub dt_out_sub                ! n_sub(number of the substations);
                                ! dt_out_sub(writing time step duration for
                                ! monitoring the substations)
first_substation_ID             ! first_substation_ID=1
type_ID n_vertices              ! type_ID(substation type ID)=1
                                ! (high-voltage),2(medium-voltage),3
                                ! (low-voltage);
                                ! n_vertices(number of vertices describing
                                ! the substation)=3,4,5,6
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 ! to be omitted in case of triangles

```

```

vertex_5_x vertex_5_y vertex_5_z ! to be omitted in case of
! ! triangles/quadrilaterals
vertex_6_x vertex_6_y vertex_6_z ! to be omitted in case of
! ! triangles/quadrilaterals/pentagons
! ! vertices of the first section (
! ! anti-clockwise viewing the substation from
! ! the outside of the fluid domain:
! ! clockwise in plan view)
... ! other possible section records
##### end substations #####

```

Figure 13.1. Commented template of the main input file of SPHERA v.9.0.0.

13.3. Tutorials

SPHERA v.9.0.0 is validated on 34 test cases (following sub-sections), each one having possible variants. Some of the tutorials 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.

13.3.1. “*body_body_impacts*”

This tutorial is completely described in Amicarelli et al. (2015). The paper version available on ResearchGate might help in case the published version is unavailable.

13.3.2. “*body_boundary_impacts*”

This tutorial is completely described in Amicarelli et al. (2015). The paper version available on ResearchGate might help in case the published version is unavailable.

13.3.3. “*dam_breach_ICOLD_trunks*”

This tutorial is described in Amicarelli & Agate (2014, SPHERIC) and Amicarelli & Agate (2015, SPHERIC). Here, the tutorial is represented by means of the mixture model for dense granular flows of Amicarelli et al. (2017).

13.3.4. “*db_2bodies*”

This tutorial is completely described in Amicarelli et al. (2015). The paper version available on ResearchGate might help in case the published version is unavailable.

13.3.5. “*db_2D_demo*”

This is a very simple and very fast tutorial representing a 2D dam break (over a flat bottom) at a rough spatial resolution. Amicarelli et al. (2013, RdS) completely describes this test case. This project report is Open-Access and also includes a synthetic English version.

13.3.6. “*db_Alpe_Gera*”

This tutorial is completely described in Amicarelli & Agate (2017). This project report is Open-Access and also includes a synthetic English version.

13.3.7. “*db_Alpe_Gera_Lanzada_substations*”

This tutorial is completely described in Amicarelli (2018). This project report is Open-Access and also includes a synthetic English version.

13.3.8. “*db_body_exp_UniBas*”

This tutorial is completely described in Amicarelli et al. (2015). The paper version available on ResearchGate might help in case the published version is unavailable.

13.3.9. “*db_ICOLD*”

This tutorial is described in Amicarelli & Agate (2014, SPHERIC).

13.3.10. “*db_multi_body*”

This tutorial is completely described in Amicarelli et al. (2015). The paper version available on ResearchGate might help in case the published version is unavailable.

13.3.11. “*db_squat_obstacle*”

This tutorial is described in Amicarelli et al. (2013). The paper version available on ResearchGate might help in case the published version is unavailable. Here, the surface mesh is generated by

SnappyHexMesh (2018) and SPHERA source code improvements for the DB-SPH treatment (Amicarelli et al., 2015, SPHERIC) apply.

13.3.12. “db_tall_obstacle”

This tutorial is described in Amicarelli et al. (2013). The paper version available on ResearchGate might help in case the published version is unavailable. Here, the surface mesh is generated by SnappyHexMesh (2018) and SPHERA source code improvements for the DB-SPH treatment (Amicarelli et al., 2015, SPHERIC) apply.

13.3.13. “dike_breach_2D_expSchHag12JHR_ID22”

This tutorial is completely described in Amicarelli & Agate (2017, RdS). This project report is Open-Access and also includes a synthetic English version.

13.3.14. “edb_2D_demo”

This tutorial of SPHERA v.9.0.0 (2015) was used as a reference test case to carry out a sensitivity analysis in Manenti et al. (2018). This paper is Open-Access and also describes in detail the present test case.

13.3.15. “edb_2D_FraCap02”

This tutorial is described in Amicarelli et al. (2017). The paper version available on ResearchGate might help in case the published version is unavailable.

13.3.16. “edb_2D_FraCap02_Taipei”

This tutorial is described in Amicarelli et al. (2017). The paper version available on ResearchGate might help in case the published version is unavailable.

13.3.17. “edb_2D_Spi05”

This tutorial is described in Amicarelli et al. (2017). The paper version available on ResearchGate might help in case the published version is unavailable.

13.3.18. “edb_ICOLD”

This tutorial is described in Amicarelli et al. (2017). The paper version available on ResearchGate might help in case the published version is unavailable.

13.3.19. “edb_KarlSand”

This tutorial is described in Amicarelli et al. (2017). The paper version available on ResearchGate might help in case the published version is unavailable.

13.3.20. “edb_Pon10”

This tutorial is described in Amicarelli et al. (2017). The paper version available on ResearchGate might help in case the published version is unavailable.

13.3.21. “floating_cube_stability”

This is a very simple and demonstrative tutorial: a solid cube is leaned on still water (rough resolution). Amicarelli et al. (2014, SPHERIC) completely describes this test case, which is also available on Amicarelli et al. (2014, RdS). This project report is Open-Access and also includes a synthetic English version.

13.3.22. “flushing_2D”

This tutorial is described in Amicarelli & Agate (2014, SPHERIC) and Amicarelli & Agate (2015, SPHERIC). Here, the tutorial is represented by means of the mixture model for dense granular flows of Amicarelli et al. (2017).

13.3.23. “flushing_3D”

As a 3D version of the previous tutorial, this test case is available on Amicarelli et al. (2014, RdS). This project report is Open-Access and also includes a synthetic English version.

13.3.24. “jet_plate”

The 7 variants of this tutorial are described in Amicarelli et al. (2013) and Amicarelli et al. (2015). Here the DBSPH boundary scheme is treated with automated procedures (Amicarelli & Agate, 2015, SPHERIC) and the scheme for body transport is corrected in terms of no-slip conditions for fluid-body interactions (Amicarelli, 2018, SPHERIC).

13.3.25. “rectangular_side_weir_Fr_0_491”

This tutorial is described in Amicarelli (2018, SPHERIC) and is also available in Amicarelli & Agate (2017, RdS). This project report is Open-Access and also includes a synthetic English version.

13.3.26. “*San_Fernando_Lower_van_Norman_dam_liquefaction*”

This tutorial is described in Amicarelli (2016, SPHERIC). Here, the tutorial is represented by means of the mixture model for dense granular flows of Amicarelli et al. (2017).

13.3.27. “*sloshing_tank_TbyTn_0_78*”

This tutorial is described in Amicarelli et al. (2013). The paper version available on ResearchGate might help in case the published version is unavailable. Here, the surface mesh is generated by SnappyHexMesh (2018) and SPHERA source code improvements for the DB-SPH treatment (Amicarelli et al., 2015, SPHERIC) apply.

13.3.28. “*sloshing_tank_TbyTn_1_07*”

This tutorial is described in Amicarelli et al. (2013). The paper version available on ResearchGate might help in case the published version is unavailable. Here, the surface mesh is generated by SnappyHexMesh (2018) and SPHERA source code improvements for the DB-SPH treatment (Amicarelli et al., 2015, SPHERIC) apply.

13.3.29. “*spherical_Couette_flows*”

This tutorial is described in Amicarelli (2016, SPHERIC). The tutorial is also available on Amicarelli & Agate (2016, RdS). This project report is Open-Access and also includes a synthetic English version.

13.3.30. “*SPH_udb_exp_Kim2015HYDROL*”

This tutorial is described in Amicarelli (2018, SPHERIC). The tutorial is also available on Amicarelli (2017, RdS). This project report is Open-Access and also includes a synthetic English version.

13.3.31. “*submerged_landslide*”

This tutorial is described in Amicarelli & Agate (2014, SPHERIC). Here, the tutorial is represented by means of the mixture model for dense granular flows of Amicarelli et al. (2017).

13.3.32. “*still_water_tank*”

This tutorial is a very simple and rough-resolution 2D test case on hydrostatic conditions.

13.3.33. “*wave_motion_for_WaveSAX*”

This tutorial is described in Peviani et al. (2017, RdS). This project report is Open-Access and also includes a synthetic English version.

13.3.34. “*wedges_falls_on_still_water*”

This tutorial is completely described in Amicarelli et al. (2015). The paper version available on ResearchGate might help in case the published version is unavailable.

14. SPHERA ACKNOWLEDGMENTS

SPHERA has been 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.
- ✓ under the Contract Agreement between RSE SpA and the Italian Ministry of Economic Development for the RdS period 2015-2017, in compliance with the Decree of 21 April 2016. Reference project: ‘A.5 - Sicurezza e vulnerabilità del sistema elettrico’, Frigerio A. et al., 2015–2018.

“We acknowledge the CINECA award under the ISCRA initiative, for the availability of High Performance Computing resources and support.” In fact, SPHERA validation has also been financed by means of the following instrumental funding HPC projects:

- ✓ HPCEFM7b - High Performance Computing for Environmental Fluid Mechanics 2017b (Italian National HPC Research Project); instrumental funding based on competitive calls (ISCRA-C project at CINECA, Italy); 2017–2018; Amicarelli A. (Principal Investigator, 160,000 core-hours), G. Pirovano.
- ✓ HPCEFM17 - High Performance Computing for Environmental Fluid Mechanics 2017 (Italian National HPC Research Project); instrumental funding based on competitive calls (ISCRA-C project at CINECA, Italy); 2016–2017; Amicarelli A. (Principal Investigator, 200,000 core-hours), G. Agate, N. Pepe, G. Pirovano, M. Torresi, B. Fortunato, F. Fornarelli, F. Scarpetta, G. Viccione, E. Pugliese Carratelli, V. Bovolin, S. Sibilla, E. Persi, G. Petaccia.
- ✓ PANCIA - High Performance Computing for Interface and Atmospheric flows (Italian National HPC Research Project); instrumental funding based on competitive calls (ISCRA-C project at CINECA, Italy); 2016–2017; Vacondio R. (Principal Investigator), A. Amicarelli, G. Curci, A. dal Palù, S. Falasca, A. Ferrari, L.M. Baldini; 110,000 core-hours.
- ✓ NMTFEPRA - Numerical Modelling of Turbulent Flows for Environment Protection and Risk Assessment (Italian National HPC Research Project); instrumental funding based on competitive calls (ISCRA-C project at CINECA, Italy); 2016-2017; Ferrero E. (Principal Investigator), A. Bisignano, A. Amicarelli, G. Curci, S. Manenti, S. Todeschini, A. Bisignano, S. Falasca. 40,000 core-hours.
- ✓ HPCEFM16 - High Performance Computing for Environmental FluidMechanics 2016 (Italian National HPC Research Project); instrumental funding based on competitive calls (ISCRA-C project at CINECA, Italy); 2016; Amicarelli A. (Principal Investigator, 146,000 corehours), G. Curci, S. Falasca, E. Ferrero, A. Bisignano, G. Leuzzi, P. Monti, F. Catalano, S. Sibilla, E. Persi, G. Petaccia.
- ✓ 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.

- ✓ 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;

The release of the FOSS versions of SPHERA have been supported and promoted by the RSE Department Director Michele de Nigris (during the period 2015-2018) and the RSE Research Team Managers Guido Pirovano (during the period 2016-2018) and Massimo Meghella (during the period 2015-2016).

15. SPHERA REGISTRATION

SPHERA v.8.0 Copyright is registered (“Registro pubblico speciale per i programmi per elaboratore, SIAE”, Italy). Since SPHERA v.8.0, the code has been developed on a public GitHub repository.

16. 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., 2018; Modellazione fluidodinamica SPH 3D per la propagazione di inondazioni in ambiente urbano e valutazioni di supporto ai fini della gestione del sistema elettrico in aree soggette a rischio idrogeologico; RSE SpA, Ricerca di Sistema, Deliverable 18001519.
4. 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 - ERCOFTAC Conference, 23-30; Parma (Italy).
5. Amicarelli A., G. Agate, 2014; SPH modelling of granular flows, 9th SPHERIC ERCOFTAC International ERCOFTAC Workshop:17-24, Paris
6. 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.
7. Amicarelli A.; G. Agate, B. Stefanova, S. Sibilla, J. Grabe; 2016; A SPH model for dike overtopping and dam liquefaction with bed-load transport, bottom drag and mobile boundaries; 11th SPHERIC ERCOFTAC International Workshop SPHERIC 2016 – ERCOFTAC Conference, pp.424-431, Garching (Munich, Germany); ISBN: 9783000533587.
8. 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
9. Amicarelli A., B. Kocak, S. Sibilla, J. Grabe; 2017; A 3D Smoothed Particle Hydrodynamics model for erosional dam-break floods; *International Journal of Computational Fluid Dynamics*, 31(10):413-434; DOI 10.1080/10618562.2017.1422731 ; JCR Impact Factor: 0.983.
10. 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.
11. 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.
12. 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.
13. Antoci C, Gallati M, Sibilla S.; 2007; Numerical simulation of fluid-structure interaction by SPH; *Computers & Structures*, 85(11-14):879-890.
14. 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.
15. 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
16. Armstrong L.M., S. Gu, K.H. Luo; 2010; Study of wall-to-bed heat transfer in a bubbling fluidised bed using the kinetic theory of granular flow; *International Journal of Heat and Mass Transfer*, 53(21-22):4949-4959.
17. 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.
18. 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.
19. 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.
20. 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
21. Berzi D., F.C. Bossi, E. Larcan; 2012; Collapse of granular-liquid mixtures over rigid, inclined beds; *PHYSICAL REVIEW E* 85, 051308:1-5.
22. 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.
23. Berzi D., J.T. Jenkins; 2011; Surface flows of inelastic spheres; *PHYSICS OF FLUIDS* 23, 013303:1-8.
24. Bonelli S., D. Marot; 2010; Micromechanical modeling of internal erosion; *European Journal of Environmental and Civil Engineering*, 15(8): 1207-1224.
25. 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.
26. Bonet J., M. X. Rodriguez-Paz; “Hamiltonian formulation of the variable-h SPH equations”; *Journal of Computational Physics*, 209 (2005), 541–558.

27. 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.
28. 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
29. 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.
30. 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
31. 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
32. 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
33. 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.
34. 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.
35. Chow M., L. Taylor, M. Chow; 1996; Time of outage restoration analysis in distribution systems; *IEEE Transactions on Power Delivery*. 11(3):1652-1658.
36. Colagrossi A, Landrini M.; 2003; Numerical simulation of interfacial flows by smoothed particle Hydrodynamics; *Journal of Computational Physics*, 191(2):448-475.
37. Colombo P., F. Coleselli; 1996; *Elementi di geotecnica*:1-500; Zanichelli.
38. Costabile P.; 2006; Propagazione di onde di dam break su fondo erodibile; XXX Convegno di Idraulica e Costruzioni Idrauliche (IDRA).
39. Crawford M., S. Seidel; 2013; Weathering the storm: building business resilience to climate change. Center for climate and energy solutions: Arlington, VA.
40. DEM2xyz v.1.0 (RSE SpA); Amicarelli A.; FOSS; <https://github.com/AndreaAmicarelliRSE/DEM2xyz>
41. 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).
42. 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.
43. Dey S.; 1999; Sediment threshold; *Applied Mathematical Modelling*, 23: 399-417.
44. 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
45. Dilts G.A.; “Moving-least-squares-particle hydrodynamics – I. Consistency and stability”; *International Journal for Numerical Methods in Engineering*; (1999), 44, 1115-1155.
46. 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.
47. 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.
48. Engage Digitizer (Mitchell et al.), <https://github.com/markummittchell/engage-digitizer>
49. 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.
50. 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.
51. 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
52. 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.
53. 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.
54. 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.
55. Feldman J. and J. Bonet; 2007; Dynamic refinement and boundary contact forces in SPH with applications in fluid flow problems; *Int. J. Numer. Meth. Engrg*; 72,295–324.
56. 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.

57. 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.
58. 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.
59. 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.
60. Fraccarollo L., Capart H.; 2002; Riemann wave description of erosional dam-break flows; *J. Fluid Mech.*, 461, 183-228.
61. 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.
62. Free Software Foundation <http://www.fsf.org/>
63. 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.
64. Gallati M., G. Braschi, Simulazione lagrangiana di flussi con superficie libera in problemi di idraulica; 2000; in "L'ACQUA 5".
65. GDAL (OSGEO); <https://github.com/OSGeo/gdal>
66. gdb (GNU, Free Software Foundation), <https://www.gnu.org/software/gdb/>
67. gedit (GNOME Foundation), <https://wiki.gnome.org/Apps/Gedit>
68. GEO-SLOPE International Ltd; The Lower San Fernando Dam; QUAKE/W Example File: Lower SanFernando Dam.doc; www.geo-slope.com
69. Git (Torvalds et al.), <https://github.com/git/git>
70. GitHub; GitHub Inc.; www.github.com
71. gfortran (GNU, Free Software Foundation), <http://gcc.gnu.org/fortran/>
72. GNU Make (GNU, Free Software Foundation), <https://www.gnu.org/software/make/>
73. Gnuplot (Williams & Kelley), <http://www.gnuplot.info/>
74. 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.
75. Gotoh H., T. Sakai; 2006; Key issues in the particle method for computation of wave breaking; *Coastal Engineering*, 53:171-179.
76. gprof (GNU, Free Software Foundation), <http://sourceware.org/binutils/docs/gprof/>
77. 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.
78. 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.
79. Grid Interpolator v.1.0 (RSE SpA); Amicarelli A.; FOSS; https://github.com/AndreaAmicarelliRSE/Grid_Interpolator
80. GSView (Ghostgum Software Pty Ltd), <https://www.ghostscript.com/>
81. Gudehus G., R.O. Cudmani, A.B. Liberos-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
82. Guzzetti F.; 2015; Frane e alluvioni, una lunga storia italiana; *ECOSCIENZA*, 3:12-13.
83. 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.
84. 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.
85. 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.
86. Hazus-MH; 2011; Technical Manual, Multi-hazard Loss Estimation Methodology, Flood Model; Department of Homeland Security, Federal Emergency Management Agency, Mitigation Division, Washington, D.C.
87. 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.
88. Holmes M.; 2015; Water infrastructure vulnerability due to dependency on third party infrastructure sectors; School of Civil Engineering and Geosciences, Newcastle University, May 2015; Doctorate thesis.
89. ICOLD - Committee on Computational Aspects of Analysis and Design of Dams; 2013; 12th Benchmark Workshop on Numerical Analysis of Dams (Graz, Austria), Theme C.
90. Image Magick (ImageMagick Studio LLC), <https://www.imagemagick.org>
91. Intel Corporation, www.intel.com

92. Inutsuka S.; 2002; Reformulation of smoothed particle hydrodynamics with Riemann solver; *Journal of Computational Physics*, 179, 238–267.
93. Johnson G.R., S.R. Beissel; “Normalized Smoothing functions for impact computations”; *Int. Jour. Num. Methods Eng.* 1996, 39: 2725-2741.
94. Kajtar J.B.; J. J. Monaghan; 2010; On the dynamics of swimming linked bodies. *European Journal of Mechanics B/Fluids*, 29:377-386.
95. 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.
96. 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
97. 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.
98. 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.
99. 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.
100. 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
101. Kramer S.L.; 1996; *Geotechnical Earthquake Engineering*:1- 653; Prentice Hall.
102. 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.
103. Kumaran V.; 2015; Kinetic theory for sheared granular flows; *C. R. Physique* 16:51-61.
104. 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.
105. 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.
106. Lambe T.W., R.V. Whitman; 1979; *Soil mechanics*:1-553; Wiley.
107. Landau L.D., E.M. Lifshitz; 1959; *Fluid Mechanics (Volume 6 of A Course of Theoretical Physics)*; 1-551.
108. Lastiwka M., N. Quinlan, M. Basa; “Adaptive particle distribution for Smoothed Particle Hydrodynamics”; *Int. J. Numer. Meth. Fluids*, 47(2005), 1403–1409.
109. 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.
110. 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.
111. Leonardi M., T. Rung; 2013; SPH Modelling of Bed Erosion for Water/Soil-Interaction; 8th SPHERIC ERCOFTAC International Workshop, 139-144, Trondheim (Norway).
112. 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.
113. 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
114. Liu M.B., G.R. Liu; 2006; Restoring particle consistency in smoothed particle hydrodynamics; *Applied Numerical Mathematics*, 56:19–36.
115. Liu MB, Liu GR.; 2010; Smoothed Particle Hydrodynamics (SPH): an Overview and Recent Developments; *Arch Comput Methods Eng*, 17:25–76.
116. 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.
117. Liu W.K., S. Jun, Y. F. Zhang; 1995; Reproducing Kernel Particle Methods; *International Journal for Numerical Methods in Fluids*, 20:1081-1106.
118. 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.
119. 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.
120. 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.

121. 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.
122. Maliszewski P., C. Perrings; 2012; Factors in the resilience of electrical power distribution infrastructures; *Applied Geography*, 32:668-679.
123. 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.
124. 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).
125. 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.
126. 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.
127. Marrone S., M. Antuono, A. Colagrossi, G. Colicchio, D. Le Touzé, G. Graziani; 2011; \square -SPH model for simulating violent impact flows; *Comput. Methods Appl. Mech. Engrg.*, 200:1526-1542.
128. 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.
129. 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.
130. 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.
131. Mayrhofer A., B.D. Rogers, D. Violeau, M. Ferrand; 2013; Investigation of wall bounded flows using SPH and the unified semi-analytical wall boundary conditions; *Computer Physics Communications*, 184: 2515-2527.
132. Mayrhofer A, Rogers BD, Violeau D, Ferrand M.; 2013; Investigation of wall bounded flows using SPH and the unified semi-analytical wall boundary conditions. *Computer Physics Communications* 184, 2515-2527.
133. Ministero dell'Ambiente e della Tutela del Territorio e del Mare (MATTM); 2018; Infrastrutture elettriche presenti sul territorio italiano; <http://sinva.ancitel.it/mapviewer/index.html?collection=http://sinva.ancitel.it/WMC/Collection/VA/D5DB339E-DB14-9144-B043-B141DCABC108&context=http://sinva.ancitel.it/WMC/Context/VA/D5DB339E-DB14-9144-B043-B141DCABC108/963AD4B4-906A-4E53-B7FC-F497C3997138&v=full>
134. 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.
135. 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.
136. Monaghan J. J.; 1992; Smoothed Particle Hydrodynamics; *Annu. Rev. Astron. Astrophys*, 30, 543-74.
137. Monaghan J.J. 1994; Simulating Free Surface Flows with SPH; *Journal of Computational Physics*, 110, 399-406.
138. Monaghan JJ. Smoothed particle hydrodynamics; 2005; *Rep. Prog. Phys.*, 68:1703-1759.
139. Monaghan J.J., J.C. Lattanzio; 1985; A refined method for astrophysical problems. *Astronomy and Astrophysics*, 149, 135-143.
140. Monaghan J.J., J.B. Kajtar; 2009; SPH particle boundary forces for arbitrary boundaries. *Computer Physics Communications*, 180:1811-1820.
141. Monaghan J.J., A. Kos, N. Issa; 2003; Fluid motion generated by impact; *Journal of Waterway, Port, Coastal and Ocean Engineering*, 129:250-259.
142. 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.
143. Morrison F. A.; 2013; *An Introduction to Fluid Mechanics*, Cambridge University Press, New York.
144. 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.
145. 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.
146. 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.
147. 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.
148. Niemunis, A., Herle I.; 1997; Hypoplastic model for cohesionless soils with elastic strain range, *Mechanics of Cohesive-Frictional Materials*, 2 (4), pp. 279-299.
149. 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

150. 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.
151. Oger G., M. Doring, B. Alessandrini, P. Ferrant; 2006; Impulse-based rigid body interaction in SPH; *Journal of Computational Physics*, 213:803–822.
152. 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.
153. OpenFOAM (OpenCFD Ltd), <https://github.com/OpenFOAM/OpenFOAM-dev>, <https://github.com/isoAdvector/isoAdvector>
154. Paraview (Kitware), <https://github.com/Kitware/ParaView>
155. 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).
156. 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.
157. 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.
158. 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.
159. Peralta C., A. Melatos, M. Giacobello, A. Ooi; 2008; Superfluid spherical Couette flow; *Journal of Fluid Mechanics*, 609:221-274; DOI: 10.1017/S002211200800236X
160. ply2SPHERA_perimeter v.1.0 (RSE SpA); Amicarelli A.; FOSS; https://github.com/AndreaAmicarelliRSE/ply2SPHERA_perimeter
161. Pontillo M.; 2010; Trasporto ed “entrainment” di sedimenti in alvei mobili; PhD thesis, Università degli studi di Napoli Federico II.
162. Price, D.J.; 2012; Smoothed Particle Hydrodynamics and Magnetohydrodynamics; *J. Comp. Phys.*, 231(3): 759-794.
163. 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.
164. 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.
165. Progetto AVI - Aree Vulnerate Italiane; CNR; 1999-2016; <http://sici.irpi.cnr.it/avi.htm>
166. Quinlan N. J., M. Basa, M. Lastiwka; “Truncation error in mesh-free particle methods”; *Int. J. Numer. Meth. Engng*, 66 (2006), 2064–2085.
167. Randles R. W., L.D. Libertsy; 1996; Smoothed Particle Hydrodynamics. Some recent improvements and applications; *Comput. Methods Appl. Mech. Engrg.*, 139:375-408.
168. Reed D.; 2008; Electric utility distribution analysis for extreme winds; *Journal of wind engineering and industrial aerodynamics*, 96(1):123-140.
169. Regazzoni P.-L., D. Marot; 2013; A comparative analysis of interface erosion tests; *Nat Hazards*, 67:937–950. DOI 10.1007/s11069-013-0620-3
170. Rexer M., C. Hirt; 2014; Comparison of free high resolution digital elevation data sets (ASTER GDEM2, SRTM v2.1/v4.1) and validation against accurate heights from the Australian National Gravity Database, *Australian Journal of Earth Sciences*, 61(2):213-226, DOI: 10.1080/08120099.2014.884983
171. Rzedkiewicz, 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.
172. 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
173. Scalasca (Forschungszentrum Jülich & TU Darmstadt), <http://www.scalasca.org/>
174. Schaeffer D.G.; 1987; Instability in the Evolution Equations Describing Incompressible Granular Flow; *Journal of Differential Equations*, 66:19-50.
175. 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.
176. 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.
177. 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.
178. Seungtaik O., K. Younghee, R. Byung-Seok; 2009; Impulse-based Boundary Force (IBF). *Computer, animation and virtual worlds*, 20:215–224.
179. 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.
180. Shepard D.; 1968; A two-dimensional interpolation function for irregularly spaced points. *Proceedings of A.C.M National Conference*, 517–524, New York (USA).

181. Shields A.; 1936; Application of similarity principles and turbulence research to bed-load movement; Ph.D. dissertation, Institute of Technology, Berlin (in German).
182. 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
183. 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.
184. “SPHERA v.9.0.0” (RSE SpA), <https://github.com/AndreaAmicarelliRSE/SPHERA>
185. 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.
186. 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.
187. Soares-Frazão 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.
188. 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.
189. 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.
190. 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.
191. 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.
192. SRTM3/DTED1 (USGS); <http://earthexplorer.usgs.gov/>
193. 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.
194. 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.
195. 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
196. 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
197. Terzaghi, K.; 1943; Theoretical soil mechanics; New York, London: Wiley.
198. 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.
199. TotalView (Rogue Wave Software), <https://www.roguewave.com/products-services/totalview>
200. 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.
201. USGS; 2012; The Los Angeles dam story; <http://earthquake.usgs.gov/learn/publications/la-damstory/>
202. 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>
203. 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.
204. Valgrind (Valgrind Developers), <http://valgrind.org/>
205. 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.
206. 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.
207. Van Rijn L.C.; 1982; The prediction of Bed-Forms and Alluvial Roughness; Report, Delft Hydraulics laboratory; The Netherlands.
208. Van Rijn L.C.; 1993; Principles of sediment transport in rivers, estuaries, and coastal seas; Aqua Publications.
209. Vaughan G.L.; 2009; The SPH equations for fluids. *International Journal for Numerical Methods in Engineering*, 79:1392–1418.
210. Vila J.P.; 1999; On particle weighted methods and Smooth Particle Hydrodynamics; *Mathematical Models and Methods in Applied Sciences*, 9(2):161-209.
211. Violeau D.; 2012; Fluid mechanics and the SPH method: theory and applications. Oxford University Press, Oxford.
212. Violeau D.; A. Leroy; 2014; On the maximum time step in weakly compressible SPH; *Journal of Computational Physics*, 256: 388-415.
213. Virtual Dub (Avery Lee), <http://www.virtualdub.org/>

214. 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.
215. 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.
216. 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
217. Wendland H., “Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree”, *Adv. Comput. Math.* 1995, 4: 389-396.
218. 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.
219. Zoppé B.; 2004; Simulation numérique et analyse de l'écoulement dans les augets des turbines Pelton; PhD thesis; Institut national polytechnique (Grenoble, France).

17. GNU FREE DOCUMENTATION LICENSE

GNU Free Documentation License
Version 1.3, 3 November 2008

Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

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.